Detailed Project Plan for the Development of the Integrated Water Resource Modeling System (IWRMS)

Phase III

System Prototyping, Development, and Delivery

2004–2006

King County

Department of Natural Resources and Parks
Water and Land Resources Division
Science Section
King Street Center, KSC-NR-0600
201 South Jackson Street, Suite 600
Seattle, WA 98104
206-296-6519 TTY Relay: 711
dnr.metrokc.gov/wlr

Detailed Project Plan for the Development of the Integrated Water Resource Modeling System (IWRMS)

Phase III

System Prototyping, Development, and Delivery

2004–2006

Acknowledgements

Pacific Northwest National Laboratories (PNNL)
IWRMS Team Work for Others Agreement No. 40420/RE0029

Don Flynn
Justin Almquist
John Burke
Karl Castleton
Andrew Cowell
Linda Deatherage
Alex Donaldson
Julie Dunkle
Sharon Eaton
Nancy Holter
Brian Homer
Bonnie Hoopes
Ranata Johnson
Liam McGrath
Mitch Pelton
Eric Robinson
Randal Taira
Dave Thurman
Gene Whelan

King County Department of Natural Resources and Parks
Water and Land Resources Division, Science Section IWRMS Team

Jeff Burkey
Curtis Degasperi
Sue Delaat
Jon Frodge
Jim Frohoff
Ahmed Garba
Tom Georgianna
Doug Henderson
Larry Jones
Kevin Schock
Jim Simmonds

Submitted by:

King County Water and Land Resources Division
Department of Natural Resources and Parks

Alternate Formats Available

206-296-7380   TTY Relay: 711

Table of Contents

List of Figures

List of Tables

Statement of Work

Introduction

The purpose of the statement of work is to define the King County staff role and responsibilities including activities that Pacific Northwest National Laboratory (PNNL)[1] will conduct in support of the King County Department of Natural Resources and Parks (KC/DNRP) to develop and deliver the Integrated Water Resource Modeling System (IWRMS).

Background and Overview

This project is organized into three phases. Phase I of this project was performed December 1999 through May 2000. Phase II (Defining Requirements and Development of Design Options and Cost Estimates) was performed July 2000 through May 2003. The scope of work provided below describes in detail the work to be accomplished in Phase III (System Prototyping, Development, and Delivery). Phase III is organized into major tasks and associated subtasks. This scope of work assumes a November 1, 2003 start date and durations for work have been based on this assumption. The scope of work provided describes each task separately, corresponding to the task-specific cost estimate provided separately. The proposed project schedule is also included (see Appendix C, IWRMS and FRAMES Master Schedule). Final delivery of the completed system is expected to occur in calendar year 2005. Including post-delivery system support the estimated period of performance for the entire project is estimated to be through calendar year 2006.

Work Breakdown

PNNL will complete the IWRMS Version 1.0 and deliver it to the KC/DNRP. PNNL will use an iterative software development process throughout the entire project. This will include delivering interim builds of the software prior to the release of the IWRMS Version 1.0 to ensure that the users have adequate input into the development of the software. To ensure that a quality product is delivered, a project and development management approach will be employed that includes project management, requirements maintenance, configuration management, integration and system testing, training, and maintenance and support of interim builds and the final release.

The scope of development activities is detailed in the following sections by task. All development tasks and estimates include development support activities mentioned above, maintenance of the design, and unit level testing of the components being developed.

PNNL will perform the following tasks to support the detailed design, development, testing, deployment, and support of the IWRMS:

[1] A table of acronyms is provided at the end of this document.

Task 1 – FRAMES Baseline Integration.
Task 2 – External Data Harvester Development.
Task 3 – Model Integration Wizard Development.
Task 4 – Data Repository and Data Pedigree Development.
Task 5 – Analysis/Visualization Tool Integration.
Task 6 – Study Manager and KC User Interface Development.
Task 7 – Computing Architecture
Project/Development Management Approach
Task 8 – Long-term Support
Task 9 – Project Oversight
Task 10 – Implementation Support
Appendix A – Configuration Management Plan
Appendix B – IWRMS and FRAMES Future Upgrades and Enhancements
Appendix C – IWRMS and FRAMES Master Schedule

These tasks are detailed in the following sections. These sections describe the scope and deliverables associated with each task. Project milestones are presented following the task descriptions as part of the project management discussion.

Task 1 – FRAMES Baseline Integration

Purpose

This section outlines the tasks needed to develop the IWRMS with Pacific Northwest National Laboratory's (PNNL's) Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES) as the underlying model integration component. The task consists of enhancements to the FRAMES 2.0 software package with any required modifications to support the County-designated models and functionality (see Appendix B for additional information on FRAMES and the IWRMS relationship). Activities within this task include the evaluation of the current version of FRAMES and all modifications of FRAMES required to meet KC/DNRP needs, including security, data pedigrees, and use of a central data repository for results storage and retrieval.

This task will also make the minimum additions/modifications required to the existing FRAMES user interface (UI) in order to support the additional integrated functionality provided by the other King County specific options added on to FRAMES. This would include UIs that perform the following functions:

a. Capture information about the pedigree of objects (models, data sets, etc.)
b. Integrate repository functionality (store/query/retrieve objects)
c. Integrate user-level security features (define access control for new objects)
d. Link to reports (pass data directly from an integrated modeling run to a report)
e. Invoke external applications (FRAMES currently supports Excel. Other third-party application linkages would be added as part of Task 5.)
f. Link to the data transformation engine
g. Update software dynamically (automatic installation of new models/applications/ updates to the client software)
h. Support for the Data Harvester (extracting data directly from data sources as well as from the Harvester's cache)

Key Objectives for Task 1

- Provide updated detailed functional requirements associated with Task 1 to make sure that deliverables are met.
- Design and implement the modifications to FRAMES needed to support the requirements that will be delivered in the IWRMS.
- Design and implement the interfaces needed to support the requirements that will be delivered in the IWRMS.

Scope

FRAMES is a software system that provides a user-friendly platform for integrating medium specific computer models as part of a holistic approach to modeling. This approach allows models of different type (i.e., source, fate and transport, exposure, health impact, and sensitivity/uncertainty), resolution (i.e., analytical, semi-analytical, and numerical), and operating platforms to be combined as part of an integrated modeling assessment. King County (County) has a need for this type of functionality as it seeks a way to perform integrated modeling assessments associated with its water resources—specifically, watershed models, lake models, river models, and human and ecological risk models. FRAMES will provide a major component of IWRMS functionality; however, some modifications will be necessary.

In this task, modifications and enhancements to FRAMES will be made in order to support the County-designated models and functionality. Included in this task will be modifications to the FRAMES Input/Output Dynamic Link Library (IODLL) used to read, write, and validate data for all model input and output files that are passed between models in FRAMES. Currently, the IODLL operates only in a single-PC operating environment. The IODLL will be modified to create a multi-user, distributed application that transforms model results and stores data pedigree information, data sets, and other data in a central repository. This task will also include any required modifications to the IODLL to enable its interoperability with other components in the system, including the Data Transformation Engine, Data Harvester, Reporting Engine, and distributed computing infrastructure.

This task will also include modifications, as required, to the existing FRAMES user interface (UI) in order to support the IWRMS.

Technical Approach

This task is predominantly a software development project that will be completed internally. The steps to complete this task are listed below and include brief descriptions and internal deliverables.

1.1 Update Functional Requirements Specification

The existing IWRMS Functional Requirements Specification dated April 15, 2003 will be updated to make sure that all requirements have been identified and prioritized to support this task. To verify that the requirements are internally consistent with the detailed design, the document will be reviewed, updated, and provided to the client with corresponding IWRMS deliverables as defined in Task 10.

1.2 FRAMES Architecture Review

Review the current FRAMES system architecture and develop a design for modifications necessary. A design document will be produced that will outline modifications to FRAMES that are required to support its use as the model integration component in IWRMS.

1.3 FRAMES 2.0 Internal Modifications

Modifications to the internal mechanisms of FRAMES identified in the design document of Task 1.2 will be made. Specifically, modifications will be made to support storing and retrieving information from a central repository and the automatic capture and storage of pedigree information. A new, functional version of the FRAMES internal mechanisms containing the repository and pedigree support will be produced. Verification of the functionality of this new version will take place by executing existing FRAMES models through it (and validating results against the old version).

1.4 FRAMES 2.0 User Interface Modifications

Identify and implement changes to the FRAMES UI to support the functionality required by the IWRMS. Specifically, modifications required to support the central repository and the capture of pedigree information will be made. A new, functional version of the FRAMES UI will be produced. Verification of the functionality of this new version will take place by executing the existing FRAMES test plans through it (augmented with the new functionality).

1.5 Task 1 Management

Task management will include tracking progress of subtasks to make sure deliverables are met in terms of time and budget and providing monthly progress reports for the task.

Information and Actions Required from the County

The plan assumes the delivery of a draft version of each document to the County with receipt of the County's comments within 10 working days, after which PNNL will make needed revisions in the next 10 working days and issue the final document.

Project Deliverables and Milestones

Milestone	**Deliverable**	**Completion Date**
An approved and prioritized IWRMS Functional Requirements Specification for the requirements related to Task 1 development	Updated Functional Requirements	February 28, 2004
Design document for architecture to support FRAMES Integration		February 27, 2004
A document describing the IODLL modifications		February 27, 2004
Modified FRAMES system to support 6-month system	Demonstration of progress at 6-month demo	July 16, 2004
Modified FRAMES system to support 12-month system	Demonstration of progress at 12-month demo	December 17, 2004
Modified FRAMES system to support 18-month system	An updated FRAMES executable and IODLL with support for multiple users, a common data repository, and updated UI	June 30, 2005
	An installation guide that outlines the instructions for installation	June 30, 2005
	Design document for architecture to support FRAMES Integration	June 30, 2005
	A document describing the IODLL modifications	June 30, 2005

Project Schedule

Task 1 is estimated to take approximately twelve months to complete, with work starting in January 2004 and ending in December 2004. The subtasks listed above are not linearly dependent on one another and some overlap is natural. The Task Lead will keep the PNNL project manager apprised of progress on each subtask in the monthly reports.

Key Staff

Core staff involved in Task 1 will include Andrew Cowell, Kevin Dorow, Bonnie Hoopes, Mitch Pelton, Karl Castleton, and Randal Taira. PNNL will involve other members of our multi-disciplinary staff to provide consulting as needed and for documentation and testing support.

Task 2 – External Data Harvester Development

Purpose

The purpose of Task 2, External Data Harvester Development, is to develop a Data Harvester that will enable a user to extract data from pre-defined data sources (both internal and external to the KC/DNRP) and transform the data to an existing schema or to create a new database structure based on a schema the user has extracted to support its use directly by models within IWRMS.

Key Objectives for Task 2

- Design and implement the user interface, harvesting engine, and task scheduler needed to administer, schedule, and facilitate the extraction of data from identified data sources.
- Design and implement the user interfaces needed to support the requirements for mapping data from the various data sources to an existing schema or new database structure that will be delivered within IWRMS.
- Design and implement the connection information needed to execute the various mappings and return the data to be transformed (if necessary) to an existing schema or new database structure to support the requirements that will be delivered in the IWRMS.
- Connect three data sources (one of each type) to be delivered in the IWRMS.
- Provide updated detailed functional requirements associated with Task 2 to ensure that the requirements are met.

Scope

The External Data Harvester consists of a mechanism to easily define, extract, and import data sets into the IWRMS from a collection of identified data sources. Potential IWRMS users require data from multiple internal and external sources as inputs to their models. The data sources vary from external and internal databases to websites. Such data must be obtained from different sources with different frequencies (i.e., daily, annually) and data from each source may be formatted differently. The data stored in IWRMS must be queryable using SQL commands. The data source types to be supported include internal databases, internal and external web pages, external ftp, internal local data file, and XML streams. The County does not access XML data sources at this time, however, they expect to in the future. Because it is not a current or pressing need, all work related to XML data sources will be a lower priority for this task.

The development process will follow the project’s iterative approach to software development. The functionality needed for the External Data Harvester development task will be documented and prioritized. This prioritization will be used to drive the features developed in the data harvester. This functionality includes the data harvester interface and engine, interfaces to map and extract the various data sources, and the ability to schedule data retrieval from internal or external sources as identified in the Functional Requirements Specification.

Technical Approach

This task is primarily a software design/development task, followed by subsequent training of County staff on the use of the resulting software. The software development task will use the Java programming language and will leverage existing standards and technologies wherever possible. User interfaces developed for this task will adhere to the technical approach outlined in the Detailed Work Plan for Task 6. The functionality outlined below is expected to be completed for the 18-month IWRMS deliverable, but incremental functionality may be delivered (if available) in the 12-month deliverable. This technical approach applies to all External Data Harvester subtasks below.

2.1 Update Functional Requirements Specification

The existing IWRMS Functional Requirements Specification will be updated to make sure that all requirements have been identified and prioritized to support the design of the Data Harvester. To ensure that the requirements are internally consistent with the detailed design, the document will be reviewed with the County, updated as necessary, and provided as part of the requirements definition activities in Task 10.

2.2 Data Harvester Design

The Data Harvester components (extraction engine, scheduling engine, and various user interface components) will be designed, including details regarding how the Data Harvester will interact with the Data Repository (Task 4). User interface designs will include screen prototypes on which County feedback will be solicited. The design, including decisions and constraints, will be documented in a design document, which will undergo internal review for technical soundness.

2.3 Data Harvester Configuration Support

Data access interfaces and data specifications will be defined in support of the harvesting capability, requiring extensions to the existing repository data model. Where changes to the data repository are required specifically to support data harvesting activities, such activities will be funded under this task.

2.4 Data Harvester Administration Development

This subtask focuses on the development of an application, including necessary user interface components, to enable a user to administer the extraction of data from internal and external data sources. This will include, in some cases, extraction and storage of raw data to the IWRMS repository for further processing (externally). This application will provide the ability to create new harvesting processes and maintain those already in existence. Discussions with the County are needed to define and understand the extraction methods (e.g., real-time extraction for data sources that are always available and/or scheduled (batch) extraction).

Several user interface components will be developed, enabling the user to map from the various formats to an existing schema or a newly created database structure based on a schema the user has extracted. The following data source formats are expected to be supported, pending discovery and documentation of DNRP user requirements:

- HTTP Mapping UI to map data from a web page to the existing schema or newly created database structure.
- FTP / Local Data File Mapping UI to map data from a file (transferred via FTP or local file system) to the existing schema or newly created database structure. .
- SQL Mapping UI to map data from a JDBC / ODBC compliant database to an existing schema or newly created database structure.
- XML Mapping UI to map data from a XML structure to an existing schema or newly created database structure.

2.5 Data Harvester Engine Development

A Data Harvester Engine will be developed that uses data source connection and data transformation information specified in the Data Harvester Administration interface (subtask 2.4 above), executes the extraction based on the information supplied, and stores the transformed results as a data set in the IWRMS data repository. Pending confirmation via planned requirements-gathering activities (Task 2.1), such extraction capabilities will be developed to support the following formats:

- A module for HTTP extraction that connects to a web page data source, executes an HTTP mapping, and returns the data to be transformed (if necessary) to an existing schema or newly created database structure. .
- A module for FTP / Local Data File extraction that connects to an FTP data source, downloads a file, executes an FTP or local file system mapping, and returns the data so that it can be transformed (if necessary) to an existing schema or newly created database structure.
- A module for SQL extraction that connects to a SQL data source, executes a SQL mapping, and returns the data to be transformed (if necessary) to an existing schema or newly created database structure.
- A module for XML extraction that connects to an XML data source, executes an XML mapping, and returns the data to be transformed (if necessary) to an existing schema or newly created database structure.

2.6 QA/QC Database UI

The QA/QC Database UI will provide users with the ability to generate QA/QC data templates from existing raw data and an interface for capturing metadata about the QA/QC process being performed. In addition, it will allow users to indicate when QA/QC processing on a particular data set is complete (so that it can be incorporated into modeling).

2.7 Extension of FRAMES Data Owners Tool

The FRAMES Data Owners Tool (DOT) will be extended to support the mapping of finalized QA/QC databases to IWRMS data specifications. The DOT allows users to map the elements of a database schema to a data specification via a graphical user interface. It will be extended to support operation in the IWRMS environment.

2.8 Data Harvester Scheduler Development

This task includes the development of the functionality that enables a user to schedule when the Data Harvester should extract data from a specified source. The scheduler will take a predefined mapping that has been generated using the appropriate mapping UI along with the data source connection information and invoke the Data Harvester Engine to collect the appropriate data from the source. The scheduler will automatically send users email notifications of completion or failure of scheduled jobs. The task also includes the time needed to investigate existing scheduling capabilities built into Windows to provide the most efficient solution.

2.9 Connection to Data Sources

This task includes connecting three data sources: one data source for the following data types: HTTP, FTP, and SQL. PNNL will be responsible for connecting to each one of these initial data source types and will demonstrate such connections (as available) to the County as part of an incremental prototype when the Data Harvester is complete (likely the 18-month prototype demonstration). As noted earlier, XML data sources do not exist at the present time but are expected to be needed in the future. PNNL does not expect XML streams to be available before the completion of this task.

County staff will be involved in this process to verify that the data harvesting process is understood and accurately supports the County’s needs. In addition, one more data source will be connected as a joint effort with the County staff. This will serve as a feedback mechanism to PNNL, both on the process and the documentation available for data source connection. Subsequent data sources will be connected by County staff. Training for data source connection using the Data Harvester will be included in Task 10 deliverables.

2.10 User Documentation

The IWRMS user documentation will be developed to provide detailed descriptions of using the Data Harvester for the different data types and will outline the instructions for installing the Data Harvester. The documentation will be provided to the client with corresponding IWRMS deliverables as defined in Task 10.

2.11 Task 2 Management

Task management will include tracking progress of subtasks to make sure deliverables are met in terms of time and budget and providing monthly progress reports for the task.

Information and Actions Required from the County

County staff will need to provide requirements for the functionality of the Data Harvester, including specification of prototypical data sources, definition of necessary data source types, and access to internal data sources. As the detailed design is implemented, County staff will need to verify the correctness and completeness of the Data Harvester for their needs. In addition, County feedback with respect to the user interface/administration components will be critical to ensuring that the Data Harvester adequately meets the needs of County staff subsequent to final IWRMS delivery.

Project Deliverables and Milestones

Milestone	Deliverable	Completion Date
Task 2 Design Document and Review Complete (Task 2.2)		November 19, 2004
Updated functional requirements specification as it relates to the 12-month deliverable (Task 2.1)		December 9, 2004
Build and Integration Test Plan updated for Web, and Database, (Task 10)		December 17, 2004
Type Database Extractor and UI complete (Task 2.4, 2.5)		January 3, 2005
Type HTTP Extractor, and UI complete (Task 2.4, 2.5)		January 20, 2005
	External Data Harvester functionality to support 12-month demo (partial completion of tasks 2.3, 2.4, 2.5)	January 2005
Type FTP/Local Data File Extractor, and UI complete (Task 2.4, 2.5)		February 24, 2005
Data Harvester Configuration Support Complete (Task 2.3)		March 31, 2005
Type XML Extractor, and UI complete (Task 2.4, 2.5)		March 31, 2005
IWRMS connection to type Database data source complete (Task 2.9)		April 20, 2005
IWRMS connection to type HTTP data source complete (Task 2.9)		April 29, 2005
QA/QC Database UI Complete (Task 2.6)		May 2, 2005
IWRMS connection to type FTP/Local Data File data		May 10, 2005

Milestone	Deliverable	Completion Date
source complete (Task 2.9)		
Modifications to FRAMES Data Owner Tool Complete (Task 2.7)		May 16, 2005
Build and Integration Test Plan updated to include Scheduler (Task 10)		May 31, 2005
Data Harvester Scheduler complete (Task 2.8)		June 6, 2005
Tech transfer to County for one additional data source complete (Task 2.9)		June 30, 2005
	External Data Harvester functionality to support 18-month system (Tasks 2.3, 2.4, 2.5)	July 2005
Updated functional requirements specification as it relates to the 18-month deliverable (Task 2.1)		July 2005
Updated user documentation complete (Task 2.10)		September 2005
Updated installation instructions complete (Task 2.10)		September 2005
	External Data Harvester to support IWRMS Version 1.0	November 2005

Project Schedule

Task 2 is estimated to take approximately 11 months to complete, with work starting in August 2004 and ending in July 2005. Limited Task 2 work will continue for the life of the IWRMS project as necessary modifications are made and errors are addressed. The subtasks listed above are not linearly dependent on one another and some overlap is natural. The Task Lead will keep the Project Manager apprised of our progress on each subtask in monthly reports.

KEY STAFF

PNNL/Battelle staff involved in Task 2 will include Dave Thurman, Ranata Johnson, Kevin Dorow, Randal Taira, Eric Robinson, Andrew Cowell, Liam McGrath, and Sharon Eaton.

Task 3 – Model Integration Wizard Development

Purpose

The purpose of Task 3, Model Integration Wizard Development, is to provide the means to easily integrate new models into the Integrated Water Resource Modeling System (IWRMS). When complete, Task 3 will result in a set of tools that enable modelers to register models[2] by providing sample input, output, and control files and mapping critical elements within those files to system defined data specifications. In addition, the task will also provide the necessary mechanisms to use information acquired during model registration to execute the model within IWRMS.

Key Objectives for Task 3

- Update requirements specification to capture the County's needs for integrating models into the IWRMS and functionality developed within this task.
- Create data specifications that capture the information that passes between the following models: HSPF (for watersheds), CH3D for lake hydrodynamics, CE-QUAL-ICM for lake water quality, CE-QUAL-W2 (river model for Sammamish River), and a risk model (TBD).
- Design and implement the Model Integration Wizard that will be delivered in the IWRMS (including associated documentation).
- Test the Model Integration Wizard and train the King County staff on its use by integrating County models identified above. The integration of these models will provide PNNL with the opportunity to test the Model Integration Wizard functionality and train County staff on its use. The integration of any additional models will be the County's responsibility.

Scope

The Model Integration Wizard task includes the development of a mechanism to incorporate new models into the IWRMS. KC/DNRP requires that models of varying types can be incorporated into the IWRMS by the model developers such that they can be easily chained together to perform integrated modeling studies.

The development process will follow the project's iterative approach to software development.

The functionality needed to support the Model Integration Wizard development task will be documented and prioritized. This prioritization will be driven by the functionality relied upon by other components that are being concurrently developed within the IWRMS and by the Functional Requirements Specification.

[2] See Definitions section at the end of this document for a detailed definition of meaning of "model" within the context of the IWRMS.

Technical Approach

This task is primarily a software development task combined with a testing / training task that uses the resulting software. The software development task will be completed using the Java programming language and will leverage existing standards and technologies wherever possible. The steps required to complete the task have been broken down into following subtasks:

3.1 Update Functional Requirements Specification

The existing IWRMS Functional Requirements Specification will be updated to reflect the County's need for integrating models into IWRMS subsequent to final system delivery. To verify that the requirements are internally consistent with the detailed design, the document will be reviewed with the client, updated as appropriate, and delivered to the client (as part of activities defined in Task 10).

3.2 Model Integration Wizard Architecture Review and Design

Existing design work and the previously demonstrated Model Setup Prototype development activities (conducted during Phase II of this project) will be reviewed and modified as necessary to produce a detailed design. A design document will be produced that outlines the software architecture and development activities for creating the Model Integration Wizard and integrating it into the IWRMS.

3.3 Model Integration Wizard Configuration Support

Data structures and data access software will be defined in support of the model integration capability, likely requiring extensions to the existing repository data model. Where changes to the data repository are required specifically to support model integration activities, such activities will be funded under this task.

3.4 County Model Data Specification Development

Data specifications will be developed to describe the information that is consumed and/or produced by the following models: HSPF, CH3D, CE-QUAL-ICM, CE-QUAL-W2, and risk model (TBD). The specific modeling element (model or data source) to modeling element connections that will be captured within these specifications will be defined, documented and verified via a review by the County modelers. The data specifications will be developed in the form of data dictionaries that will be developed using the Dictionary Editor that is available with the FRAMES toolset. Both PNNL developers and the County modelers will be involved in the creation of these dictionaries. The County modelers will be necessary to provide their domain expertise with the specific model linkages and need to become familiar with the process of incorporating new models.

3.5 Model Registration User Interface

A user interface will be developed that can be used to acquire information about a model during the registration process. It will be the primary interface for model registration and will capture the following types of information: executable and related files (i.e., DLL's or other files that when combined form the executable), system requirements, version, calibration information, etc. In addition, it will link into the software components generated from subtasks 3.6 and 3.7 to capture the data requirements of the model. The information that is gathered from this interface will be stored in the Repository as part of the model registration.

3.6 Control File Definition User Interface

User interface components will be developed to enable the user to define how a model control file should be generated based on a sample control file provided by the user. The model control file is used to specify the operating parameters for the execution of the model. These components will give the user the ability to define the editable and non-editable parameters within the control file, identify specific areas where user-defined values are required, and describe the types of values that are appropriate in each instance.

3.7 Data File Transformation Generator User Interface

A user interface will be created that will enable the user to define how the data in the model input and output files maps to system-defined data specifications (identified in Task 3.4).

3.8 File Input / Output Module

A software component will be created that will be responsible for reading in the sample input, output, control file(s), and model executable file(s) provided during the model registration process. It will relay the appropriate parts to the other components that are used in model registration (Tasks 3.5-3.7 and 3.9-3.11). It will also be responsible for outputting these files to the model execution components (Tasks 3.12 and 3.13) for use during model runs.

3.9 Parsing Application Programming Interface (API)

A software component will be created that is responsible for reading model files (input, output, and control) and breaking them up into components such that they can be processed by the Model Integration Wizard UI's. In this task, PNL will leverage existing parsing technologies where applicable to handle converting the sample model input, output, and control files into XML representations. These XML representations can then be handed off to the components described in Tasks 3.10 and 3.11 for further processing.

3.10 Control File Definition Generator

Software components will be developed that can take the information gathered by the Control File Definition UI along with the sample control file that was submitted during registration and generate the following:

1. A template control file with all non-static data removed.
2. A definition of the non-static values in the control file that can be used to dynamically generate a user interface for capturing user supplied information about a particular modeling run (along with information on where to place that information in the template file).

Both of these elements will then be stored in the data repository for future use.

3.11 Data File Transformation Generator

Software components will be developed that can take the information gathered by the Data File Transformation Generator UI along with sample input and output files that were submitted during the registration and perform the following:

1. Copy static data files to the repository.
2. Create data transformation definitions that map the data in the model specific formats to the system defined data specifications (along with any other static information that is required in the data files).

3.12 Data Transformation Engine

A software component will be developed that can take a transformation definition and a data set file as input and perform the transformation defined by the definition on the data set and produce an output file. This will be used to (1) transform data that exists in the repository (in a data specification format) into the specific format required by the model for execution and (2) transform data that is produced by the model to a data specification format that can be stored in the Repository (and passed on to down stream models).

3.13 Model Setup Program

A software component that can use the information generated from the model registration process (subtasks 3.5, 3.6, 3.7, 3.10, and 3.11) to:

1. Generate the appropriate model input files from data that conforms to the system defined data specifications.
2. Generate a GUI to acquire any user defined values for the control file.
3. Set up the model for execution.
4. Transform the model specific outputs to data sets that conform to the system defined data specifications (using the Data Transformation Engine).

In addition, this task will included generating a FRAMES compatible preprocessor and postprocessor such that this component can be seamlessly integrated into the FRAMES model connection framework.

3.14 Integration of County Models

Upon completion of testing the integration wizard by PNNL, the following models will be integrated into the IWRMS: HSPF, CH3D, CE-QUAL-ICM, CE-QUAL-W2, and a risk model (TBD). This will be a joint effort with the PNNL developers and King County modelers as an opportunity to train County staff on the use of the Model Integration Wizard. Any additional models will be integrated entirely by County staff.

3.15 User Documentation

The IWRMS user documentation will be developed to provide detailed descriptions of using the Model Integration Wizard for integrating and executing models and will outline instructions for its installation. The documentation will be provided to the client with corresponding IWRMS deliverables as defined in Task 10.

3.16 Task 3 Management

Task management will include tracking progress of subtasks to make sure deliverables are met in terms of time and budget and providing monthly progress reports for the task.

Information and Actions Required from the County

County staff involvement will be required in the following tasks:

- Task 3.4 — In this task the County modelers will need to be involved in creating the data specifications that define the data that interchanges between the different model types. PNNL development team members will need to meet with the County modelers for a one day meeting to define the specifications, along with additional feedback (via phone) for any further refinements that are required.
- Tasks 3.5-3.12 — For each of the models to be integrated (HSPF, CH3D, CE-QUAL-ICM, and CE-QUAL-W2) the County model owner will be required to provide PNNL developers with (1) the model executable, (2) a calibrated set of input files with definitions of the file formats such that the sources of input data can be determined (relationship to data specifications developed in Task 3.4), (3) a valid control file with a definition of the file format such that user adjustable parameter locations and valid value ranges are specified, and (4) a sample set of output files with definitions of the file formats such that the locations of data elements related to data specifications developed in Task 3.4 are identified. In addition to the above information, the modelers will also need to be available to PNNL developers for any questions in regards to the model setup / execution process.
- Task 3.14 — This task is a joint effort between PNNL developers and the County modelers. PNNL developers will work with County modelers to integrate the models as PNNL will have an understanding of the tool and the modelers will have the understanding of model file formats and parameters. This collaborative arrangement will be part of the technology transfer to the County. Once we have gone through the process together on two to three models, the County should possess enough of an understanding of how the tool operates that they can integrate the remaining models on their own. PNNL would be available to provide support to the County in their efforts. This support will be limited to the integration of the currently planned IWRMS models. The integration of models beyond the current set will be the sole responsibility of the County. We have estimated that the PNNL effort to assist in integrating the first few models will be between 16 and 40 hours. No estimate for the County time required was made but it should be in the same range or less. The work will take place at the Battelle-Seattle office or at the King County offices to the extent possible.

In addition, this task will require feedback from the County staff on an as-needed basis during the software development (which will take place via phone conversations / email when possible).

Project Deliverables and Milestones

Milestone	Deliverable	Completion Date
Design document for architecture of the Model Integration Wizard (Task 3.2)		October 1, 2004
Updated functional requirements specification as it relates to the 12-month deliverable (Task 3.1)		November 30, 2004
Data Specifications for all model types to be incorporated into the first release of the IWRMS (Task 3.4)		November 30, 2004
Generic Model User Interface completed (Task 3.5), Control File Definition User Interface completed (Task 3.6),		November 30, 2004
File Input / Output Module Completed (Task 3.8)		December 17, 2004
Parsing Application Programming Interface (API) completed (Task 3.9)		January 4, 2005
Control File Definition Generator completed (Task 3.10)		January 28, 2005
Data File Definition Generator completed (Task 3.11)		February, 18, 2005
Data Transformation Engine complete (Task 3.12)		March 11, 2005
Model Setup Program completed (Task 3.13)		April 22, 2005
King County model integration completed (Task 3.14)		July 1, 2005

Milestone	Deliverable	Completion Date
	Model Integration Wizard functionality to support 12-month demonstration	January 2005
	Model Integration Wizard functionality to support 18-month demonstration	July 2005
	Model Integration Wizard to support IWRMS Version 1.0	November 2005

Project Schedule

Task 3 is estimated to take approximately ten months to complete, with work starting in July 2004 and ending in April 2005. Limited task work will continue for the life of the project as necessary modifications are made and errors are addressed. The subtasks listed above are not linearly dependent on one another and some overlap is natural. The Task Lead will keep the Project Manager apprised of our progress on each subtask in monthly reports.

Key Staff

PNNL/Battelle staff involved in Task 3 will include Dave Thurman, Ranata Johnson, Kevin Dorow, Randal Taira, Eric Robinson, Andrew Cowell, Karl Castleton, Alex Donaldson, and Sharon Eaton.

Task 4 – Data Repository and Data Pedigree Development

Purpose

The purpose of Task 4, Data Repository and Pedigree Development, is to further design, develop, and implement the central data repository for the Integrated Water Resource Modeling System (IWRMS). When complete, Task 4 will provide a central data repository that will contain data for models, specifications, transformations, data sets (from data sources or model runs), studies, pedigrees, and applications. In addition, the task will provide capabilities for reporting and archiving data and the necessary user interfaces to administer the stored data within the data repository for all requirements delivered in Version 1 of the IWRMS.

Key Objectives for Task 4

- Provide updated detailed functional requirements that demonstrate the capabilities to be integrated in each deliverable of the IWRMS.
- Design and implement the data model needed to support the requirements that will be delivered in the IWRMS.
- Design and implement the interfaces needed to support the requirements that will be delivered in the IWRMS.

Scope

The Data Repository and Data Pedigree task consists of developing a central storage facility for data that are produced within or imported into the IWRMS. It is the data store for all elements in the system. It will contain elements to support data transformations, creation and maintenance of data specifications, study management, pedigree information, registration of models, data sets (either from data sources or model runs results), reports, registration of applications, and functionality to archive information in the data repository. It will also provide the capability of storing raw data harvested from internal and external sources and QA/QC processed data that is generated from the raw data. In addition, all data elements will be stored in a form such that they are easily queryable through the use of Structured Query Language (SQL), the common query language of databases

The development process is based on an iterative lifecycle. The functionality needed for the Data Repository and Data Pedigree development task will be implemented based on the priority of the development tasks in order to make sure that the data repository functionality is in place to support other efforts. This functionality includes other data transformations, data specifications, models, and data sets specific to the project, reporting capabilities, application registration, and archiving capabilities as identified in the Functional Requirements Specification. PNNL will develop the necessary interfaces to support the use of the data repository functionality to make sure County personnel can retrieve, import, and administer data from the data repository.

Technical Approach

This task will use a combination of a relational database (Microsoft SQL Server 2000) and a file system as the basis for the data repository. There is the possibility of using a version control system as a third component in the data repository composition. The supporting interfaces are expected to be written in Java with a data persistence layer written in C++. This task will use the file system tools available within FRAMES whenever possible. Each subtask will determine through design analysis which approach is most efficient for implementation (data model in SQL, file system or a combination of both). When SQL is used PNNL may rely on the existing browser SQL tools whenever possible. The design for the data model that defines the data repository and its elements will be captured in an Entity Relationship Diagram (ERD) and a FRAMES Object Model before implementation. The software services interface will be made available to the other parts of system via web services. This technical approach applies to all Data Repository and Data Pedigree subtasks below.

4.1 Updated Functional Requirements Specification

The existing IWRMS Functional Requirements Specification dated April 15, 2003 will be updated to makes sure that all requirements have been identified and prioritized to support this task. To verify that the requirements are internally consistent with the detailed design, the document will be reviewed, updated, and provided to the client with corresponding IWRMS deliverables as defined in Task 10.

4.2 FRAMES Data Repository and Pedigree Integration Development

Building on the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES) review in Task 1, this task will generate the design and implementation of the design needed to support moving FRAMES to the IWRMS data repository architecture. The IWRMS design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. This task also includes the creation an interface to the FRAMES 2.0 API that is needed to facilitate development across programming languages because the FRAMES 2.0 API is a C-style interface. This task will create a new Java interface for the FRAMES 2.0 API that can be used by other parts of the system. This API will also be augmented with pedigree information needed by the Study Manager and other software for IWRMS. This task will also create a mechanism for converting information from FRAMES 2.0 data persistence formats (file-based) into the IWRMS formats (queryable hybrid).

A user interface to support the basic retrieval and storage functions (e.g., Browse) will be developed to support the integration of the baseline version of FRAMES. The functionality expected to be demonstrated with the data repository development for the 12-month deliverable includes:

- Data transformations.
- Data specifications that comprise the FRAMES data dictionary.
- Pedigree data that include who did what in the system and where the data came from.
- Data from models, which include but are not limited to MEPAS and HWIR.

- Data sets that consist of the data populated to support models defined by the data dictionary.
- Necessary data needed to support the County-specific user interface.

4.3 Data Specification Manager

A Data Specification Manager will be designed and developed to support the data set definition that defines the data transferred between two model types. In other words, for a model to meet specifications it must produce the correct data (as specified by IWRMS) and that data must be in the correct format (as specified by IWRMS). The specification is how IWRMS will standardize the passing of data between models. This allows flexibility in swapping models in and out. The Data Specification Manager enables a user to maintain all specifications in the IWRMS. This task includes the design and implementation of a data model through discussions with County modelers to make sure that the data repository stores the specification in a common data format and those data specifications are understood. The Data Specification Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the basic retrieval and storage functions (e.g., Browse) will be developed. This functionality is expected to be demonstrated as part of the 12-month IWRMS deliverable.

4.4 Data Transformation Manager

A Data Transformation Manager will be designed and developed to support the data transformed from one specification to another. The Data Transformation Manager will enable a user to maintain all transformations in the IWRMS. This task will include discussions with County modelers to verify that the data repository stores the transformation specification in a common data format and that data transformations are understood. The Data Transformation Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the basic retrieval and storage functions (e.g., Browse) will be developed to support the data transformations. This functionality is expected to be demonstrated as part of the 12-month IWRMS deliverable.

4.5 Study Administration Manager

A Study Administration Manager will be designed and developed to support the Study Manager data that are collected and stored in the data repository. The Study Administration Manager enables a user to maintain all studies in the IWRMS. This task includes discussions with PNNL personnel that are involved with Task 6 to make sure that the Study Manager’s storage and retrieval needs are understood. The Study Administration Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the basic retrieval and storage functions (e.g., Browse) will be developed. This functionality is expected to be demonstrated as part of the 12-month IWRMS deliverable.

4.6 Pedigree Manager

A Pedigree Manager will be designed and developed to support the pedigree data stored in the data repository. The Pedigree Manager enables a user to maintain all pedigree data in the IWRMS. This task includes discussions with the County to determine the pedigree information to be stored in the data repository, which consists of a list of attributes that specifically identify every object or entity in the data repository. The Pedigree Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the basic retrieval and storage functions (e.g., Browse) will be developed. This functionality is expected to be completed in the 18-month IWRMS deliverable, but incremental functionality is expected to be included in the 12-month deliverable.

4.7 Model Manager

A Model Manager will be designed and developed to support the registration and maintenance of computational models in IWRMS. This task includes discussions with personnel involved in Tasks 1 and 3 to make sure that the data needed to support model integration are identified. The Model Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the basic retrieval and storage functions (e.g., Browse) will be developed. This functionality is expected to be demonstrated as part of the 12-month IWRMS deliverable.

4.8 Data Set Manager

A Data Set Manager will be designed and developed to support the basic retrieval and storage functions (e.g., Browse) to support the collection of data that are either produced or consumed by an object or entity in the system. The Data Set Manager enables a user to maintain all data sets in the IWRMS. This task includes discussions with personnel involved in Tasks 1 and 3 to make sure that the data needed to support the data sets produced are understood. The Data Set Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the administration of the stored data sets will be developed. This functionality is expected to be demonstrated as part of the 12-month IWRMS deliverable.

4.9 Report Manager

A Report Manager will be designed and developed that maintains all reports in the IWRMS. This task includes discussions with personnel involved in all tasks to make sure that the reports needed to be produced are identified. The Report Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the basic retrieval and storage functions and administration of stored reports (e.g., Browse) will be developed. This functionality is expected to be demonstrated as part of the 18-month IWRMS deliverable.

4.10 Application Manager

An Application Manager will be designed and developed that supports a software application that takes input, processes it, and produces visualization or analytical results. The Application Manager enables a user to maintain all applications in the IWRMS. This task includes discussions with personnel involved in Tasks 1 and 3 to make sure that the data needed to support the applications are understood. The Application Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the basic retrieval and storage functions and administration (e.g., Browse) will be developed. This functionality is expected to be demonstrated as part of the 18-month IWRMS deliverable.

4.11 Raw – QA/QC Data Manager

A Raw – QA/QC Data Manager will be designed and developed that maintains all the raw and QA/QC data that is stored in the IWRMS. This task includes discussions with personnel involved in Task 2 to make sure that the data needed to support the data harvesting process are identified. The Raw – QA/QC Data Manager design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the basic retrieval and storage functions and administration (e.g., Browse) will be developed. This functionality is expected to be demonstrated as part of the 18-month IWRMS deliverable.

4.12 Data Archiver

An application to enable a user to transfer inactive or unused data from within the data repository to offline storage will be developed. This task includes discussions with County personnel involved in the administration of the IWRMS to make sure that the archive capability needed to support the administration of the data repository is understood. The Data Archiver design will undergo a PNNL independent technical review to make sure that a technically sound design is implemented. A user interface to support the administration of the archived data will be developed. This functionality is expected to be demonstrated as part of the 18-month IWRMS deliverable.

4.13 Task 4 Management

Task management will include tracking progress of subtasks to make sure deliverables are met in terms of time and budget and providing monthly progress reports for the task.

Information and Actions Required from the County

The County will need to be involved in the various discussions about the requirements related to the Data Repository and Data Pedigree Development. As the data model is designed and solidified, the County will need to verify the correctness and completeness of the data being stored.

Project Deliverables and Milestones

Milestone	Deliverable	Completion Date
Design for the study administration complete (Task 4.5)		March 12, 2004
Design for the FRAMES repository development complete (Task 4.2 – 4.4)		May 28, 2004
Updated functional requirements specification as it relates to the 6-month deliverable (Task 4.1)		May 28, 2004
	Data repository system to support 6-month demo (Task 4.2)	July 28, 2004
Data Repository Administration Design Complete (Task 4.2)		November 18, 2004
FRAMES repository development complete (Task 4.2 – 4.4)		November 26, 2004
Raw QA/QC Repository Development Complete (Task 4.11)		December 16, 2004
Study administration repository development complete (Task 4.5)		December 17, 2004
Updated functional requirements specification as it relates to the 12-month deliverable (Task 4.1)		December 9, 2004
Pedigree repository development complete (Task 4.6)		January 19, 2005
Data specification repository development complete (Task 4.3)		January 19, 2005

Milestone	Deliverable	Completion Date
Data transformation repository development complete (Task 4.4)		February 1, 2005
	Data repository system to support 12-month demo	January, 2005*
Models repository development complete (Task 4.7)		February 21, 2005
Datasets repository development complete (Task 4.8)		March 23, 2005
Applications repository development complete (Task 4.10)		April 8, 2005
Reports repository development complete (Task 4.9)		April 25, 2005
Updated functional requirements specification as it relates to the 18-month deliverable (Task 4.1)		July 2005
Archiver repository development complete (Task 4.12)		May 16, 2005
Updates to the repository and data model as needed to support implemented IWRMS functionality complete		July 2005*
	Data repository system to support 18-month system	July 2005
	Data repository system to support IWRMS Version 1.0	November 2005

*The Data Repository system is expected to be completed before the 18-month IWRMS deliverable. The specific functionality needed to support the data repository may not be complete by this time. This may require minor updates to various pieces of the Data Repository once completed.

Project Schedule

Task 4 is estimated to take approximately 16 months to complete, with work starting in February 2004 and ending in May 2005 Limited task work will continue for the life of the project as necessary modifications are made and errors are addressed. The subtasks listed above are not linearly dependent on one another and some overlap is natural. The Task Lead will keep the Project Manager apprised of our progress on each subtask in monthly reports.

Key Staff

PNNL/Battelle staff involved in Task 4 will include Dave Thurman, Ranata Johnson, Kevin Dorow, Randal Taira, Eric Robinson, Don Flynn, Andrew Cowell, John Burke, Sharon Eaton, Liam McGrath, and Karl Castleton.

Task 5 – Analysis/Visualization Tool Integration

Purpose

The KC/DNRP has specific data analysis and visualization tools that it employs to view and analyze modeling results. This task will add capabilities to IWRMS to support the export of integrated modeling results to said tools, or in some cases to make the tools invokable from within IWRMS. In addition, a more intricate integration of some tools will be developed such that when modeling results are produced, the data will be automatically transformed to the format required by the tool and the tool will be initiated to generate results visualization. FRAMES currently has similar capabilities through which certain types of results can be exported automatically to Excel and plotted automatically using a script. A report generator will also be developed in this task that can provide reports on studies and objects stored in the repository. The main components of this task are as follows:

- Development of a capability to export modeling results in a format suitable for import into GIS tools of the County's choice.
- Extensions to IWRMS to allow third party tools (e.g., MatLab, TecPlot, etc…) to be integrated into the system, similar in functionality to the Model Integration Wizard. This includes developing a graphical UI to allow the user to register third party tools for use with the IWRMS. Included in this will be a method for allowing a user to define the mapping of the input data formats required by the tool to the system defined data specifications.
- A tool that will take the information provided during the registration and transform the data (using system defined data specifications) to the required input format. The application will then be invoked using this data.
- The built-in tools will be a set of predefined visualizations and data analysis functions that are identified as being necessary components of the system that can not be performed by 3rd party tools. These functions have been identified in the requirements.
- Development of a report generator tool that will allow the users to generate and publish reports about the information contained within the repository. Typical reports would include component (models/data sets/etc.) usage, study results, study progress (work flow), etc. The report generator should be able to support both web-based and RTF report generation.

Scope

The scope of this task is dependent upon the tools selected for integration. Prior to starting this task, PNNL will meet with the County to agree on the tools to be integrated in priority order within an established budget. The County shall supply the development team with valid licenses of the third party tools that they wish to be integrated.

Project Deliverables and Milestones

1. An approved and prioritized IWRMS Functional Requirements Specification for the requirements related to Task 5 development.
2. A modified version of IWRMS that allows for third-party tools to be integrated

3. A GIS data export capability in IWRMS.
4. A UI for registering third party data analysis/visualization tools with IWRMS.
5. A report generator tool.

Task 6 – Study Manager & King County User Interface Development

Purpose

The purpose of Task 6, Study Manager & King County User Interface Development, is to construct a user interface (UI) for the Integrated Water Resource Modeling System (IWRMS) to allow King County (County) staff to perform hydrological studies. First, a County-specific (meaning geographically relevant) UI shall be constructed to interface with the current Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES) 2.0 implementation. This will allow for the generation of integrated modeling runs by selecting geographic components of interest (i.e., watersheds, rivers, and lakes) from a map-based interface. Second, a Study Manager shall be implemented. This is a comprehensive UI that provides rudimentary functionality for all activities related to our hydrological studies. This includes being able to create new studies (via a New Study Wizard discussed next), view studies in progress and those that have been completed (down to each study's individual activities), edit study details as required (with study pedigree stored in a log file), and delete or archive studies that are obsolete. Third, a New Study Wizard shall allow the user to walk through the steps required to create a new study. This includes miscellaneous information describing the study (study title, description, requestor and creator, due date, etc.) as well as specific geographical and temporal constraints. Finally, the Study Workbench, while formally part of the New Study Wizard, deserves special consideration. The Study Workbench consists of a toolbar of modeling components (models and data sets) that may be used to interact with workbench canvas. This interactive tool aims to provide a simple and natural interface for different classes of users to be able to interact with a study model at a level of detail of their choosing (site, model, or component). In addition to these four task elements, connections to other features of the IWRMS system (described within the other chapters) shall be centered within the study manager.).

Key Objectives

- Develop intuitive design solutions for the numerous UI requirements.
- Implement the design solutions to produce an instinctive and natural interface to the underlying complexity of an integrated water resource modeling system.
- Perform detailed usability analysis to make sure the product is natural and easy to use.
- Provide thoughtful training and guidance for users on how to use the system to perform their tasks.

Scope

The scope of this effort will range from updating the design sketches produced in Phase II and lead all the way to client usability studies. Within the first six months, leading to a demonstration of progress in July 2004, we intend to complete a number of tasks:

1. Review the design sketches from Phase II and verify that they truly capture the presentation and interaction required to engage the client in the task. Further, we intend to confirm with the client that the UIs expressed in the design sketches are sufficient.

Before expending a great deal of time and effort implementing the UIs, it is essential for domain experts to review and approve such designs.

2. Implement a subset of the functionality as a *broad but shallow* prototype (meaning that while it will appear to contain a large coverage of the overall functionality, the majority of the complexity will be imitated). This is to provide the client with as much detail as to how the system will *look and feel* and allow the client to provide feedback. The particular elements that we intend to produce for July include the following:

    a. The Study Manager shall be implemented as a lightweight client, written as a Java application. It shall include the stubs to other functionality such as model registration, data transforms, etc. as well as any functioning components that are available at that time. We envision, using either a synthesized data model or the repository, demonstrating a set of studies in various steps of completeness. Some editing of these studies will be available and a log shall store study pedigree.

    b. The New Study Wizard, including a simplified geographic extents screen. We shall also implement a feature-limited Study Workbench separately from the New Study Wizard and integrate them together before the demonstration. The New Study Wizard will be executed by selecting **New Study** from the Study Manager menu bar. It shall consist of four steps: 1) a study information step, 2) graphical extents, 3) the Study Workbench, and 4) an activity assignment step. As step 4 is intrinsically connected to the Study Workbench, it shall be developed as part of that component. Our aim is to provide near complete functionality for one simple scenario in order to provide an environment for rich feedback from the County.

    c. The Study Workbench consists of steps 4 and 5 of the New Study Wizard, namely the Study Workbench component itself, where users can graphically map out a study at their desired level of detail, and the activity assignment step where activities are assigned to staff. We are working to have most of the functionality available for July so that the client can provide feedback about the interaction mechanisms.

Usability studies shall be conducted around the same time as the demonstration and any serious problems shall be addressed by redesign. The changes and additions to FRAMES 2.0 are described in detail in Task 1 but shall incorporate elements of the New Study Wizard that allow for geographic-specific interaction.

Technical Approach

This task will use the *Microsoft Windows 98/2000 UI Design Guidelines*[3], in conjunction with the paper prototype designs, to create a task-centered, natural environment for the user. The

[3] http://msdn.microsoft.com/library/default.asp?URL=/library/books/winguide/ch00a.htm

technical approach for each UI element follows the same format and is presented only once here, with individual details highlighted.

6.1 Update Functional Requirements Specification

The existing IWRMS Functional Requirements Specification dated April 15, 2003 will be updated to make sure that all requirements have been identified and prioritized to support this task. To verify that the requirements are internally consistent with the detailed design, the document will be reviewed, updated, and provided to the client with corresponding IWRMS deliverables as defined in Task 10.

6.2 Refine Paper Prototype Designs

The current designs, while successfully presenting the proposed functionality, shall be updated to cover current practices and to take advantage of any current code available for reuse.

6.3 Obtain Client Verification

The initial designs were presented to the client in May 2003. As we move along, it is essential that the client (and especially members of the client staff that will be tasked with using the IWRMS) are continually involved and that their feedback be taken into consideration.

6.4 Set up Development Environment

While prototype redesign and client verification is ongoing, we shall move forward in setting up an implementation environment. This includes working with team members to define tools to be used, common data formats, and underlying data model.

6.5 Implement UI

The initial development step shall be to build the graphical elements of the UI components to correspond to the design sketches. This includes writing the code that produces the UI elements of the Study Manager, New Study Wizard, and Study Workbench as well as the interaction mechanisms.

6.6 Incorporate Data

To make sure that the look and feel and the interaction match what was envisioned in the designs, certain test data shall be either produced or utilized and shall match the IWRMS entity-relation diagram.

6.7 Conduct Usability Studies

We envision three types of study. First, at regular points during the development lifecycle, we shall meet with County personnel and gather feedback that shall be incorporated back into the design. We shall also perform internal studies for UI usability and consistency (UI unit testing). PNNL staff shall run this study and perform any required redesign and implementation corrections. Finally, external usability studies shall be executed at the Seattle office, using County staff. It is envisioned that this will take place at the 18-month alpha release stage. Instead of uncovering consistency and interface problems, we foresee this step as highlighting only domain-specific issues. These shall be recorded, and any redesign and reimplementation required shall be performed.

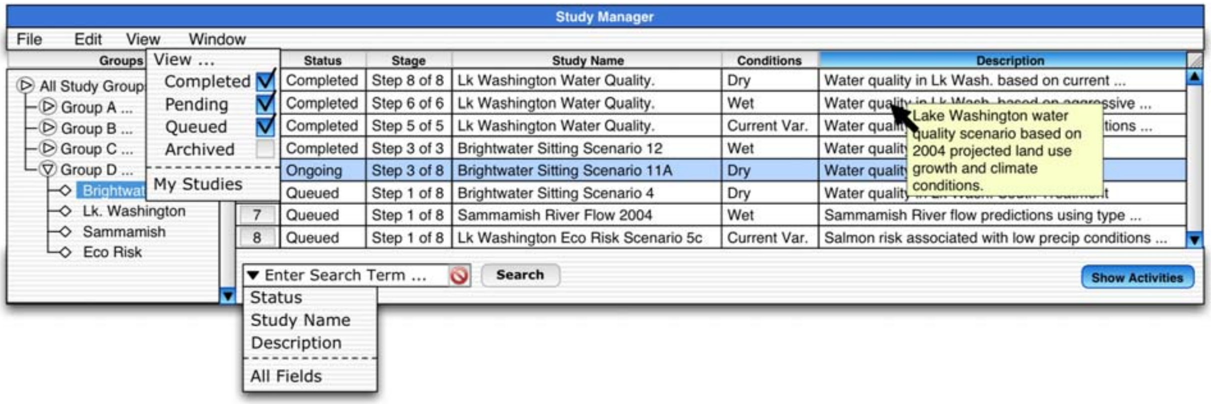
Study Manager/User Interface Development

Three additional components will be developed to support the creation, execution, and archiving of scientific modeling studies:

- A comprehensive UI to provide rudimentary functionality for study maintenance (viewing, updating and deleting of studies currently underway).
- A study builder 'wizard' that leads the user through the steps required to formulate a study.
- Interface elements that allow connection to other features offered by the system (e.g., interfaces to manage data transforms, models, and the repository).

Features of these components are illustrated in the prototype Study Manager screen shots developed in Phase II of this project:

- Study Manager Main Screen — Figure 1 illustrates the initial Study Manager interface. A list of currently active studies and information about each study is presented. This interface enables users and administrators to track active studies, identify security parameters (who can access, view, etc.), as well as providing search mechanisms for archive/retrieval purposes.

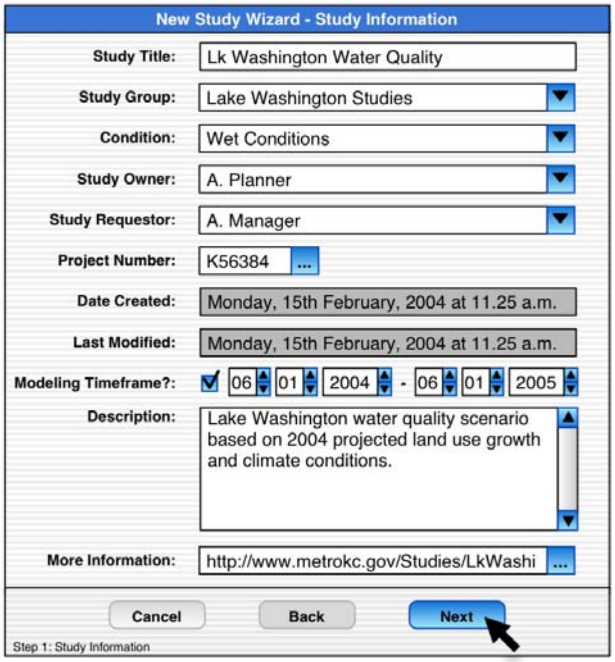


	Status	Stage	Study Name	Conditions	Description
	Completed	Step 8 of 8	Lk Washington Water Quality.	Dry	Water quality in Lk Wash. based on current ...
	Completed	Step 6 of 6	Lk Washington Water Quality.	Wet	Water quality in Lk Wash. based on aggressive ...
	Completed	Step 5 of 5	Lk Washington Water Quality.	Current Var.	Water qual ... tions ...
	Completed	Step 3 of 3	Brightwater Sitting Scenario 12	Wet	Water quality
	Ongoing	Step 3 of 8	Brightwater Sitting Scenario 11A	Dry	Water quality
	Queued	Step 1 of 8	Brightwater Sitting Scenario 4	Dry	Water quality ... t
7	Queued	Step 1 of 8	Sammamish River Flow 2004	Wet	Sammamish River flow predictions using type ...
8	Queued	Step 1 of 8	Lk Washington Eco Risk Scenario 5c	Current Var.	Salmon risk associated with low precip conditions ...

Lake Washington water quality scenario based on 2004 projected land use growth and climate conditions.

Figure 1 – The Study Manager Main Screen

- Study Builder Wizard — The following figures envision UI components that capture information to identify a study (e.g., name, description, creator, etc.; see Figure 2), specify the geographic extents of a study (Figures 3), and subsequently specify the steps or activities involved in a study by graphically constructing a workflow diagram (Figures 4 & 5).



New Study Wizard - Study Information

Study Title: Lk Washington Water Quality

Study Group: Lake Washington Studies

Condition: Wet Conditions

Study Owner: A. Planner

Study Requestor: A. Manager

Project Number: K56384

Date Created: Monday, 15th February, 2004 at 11.25 a.m.

Last Modified: Monday, 15th February, 2004 at 11.25 a.m.

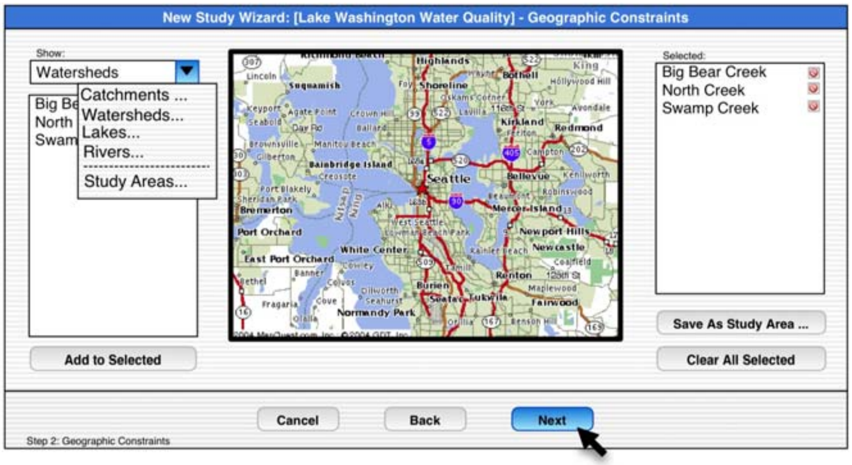
Modeling Timeframe?: ☑ 06 01 2004 - 06 01 2005

Description: Lake Washington water quality scenario based on 2004 projected land use growth and climate conditions.

More Information: http://www.metrokc.gov/Studies/LkWashi

Step 1: Study Information

Figure 2 – Study Information



New Study Wizard: [Lake Washington Water Quality] - Geographic Constraints

Show: Watersheds

Selected:
Big Bear Creek
North Creek
Swamp Creek

Step 2: Geographic Constraints

Figure 3 – Geographic Extents

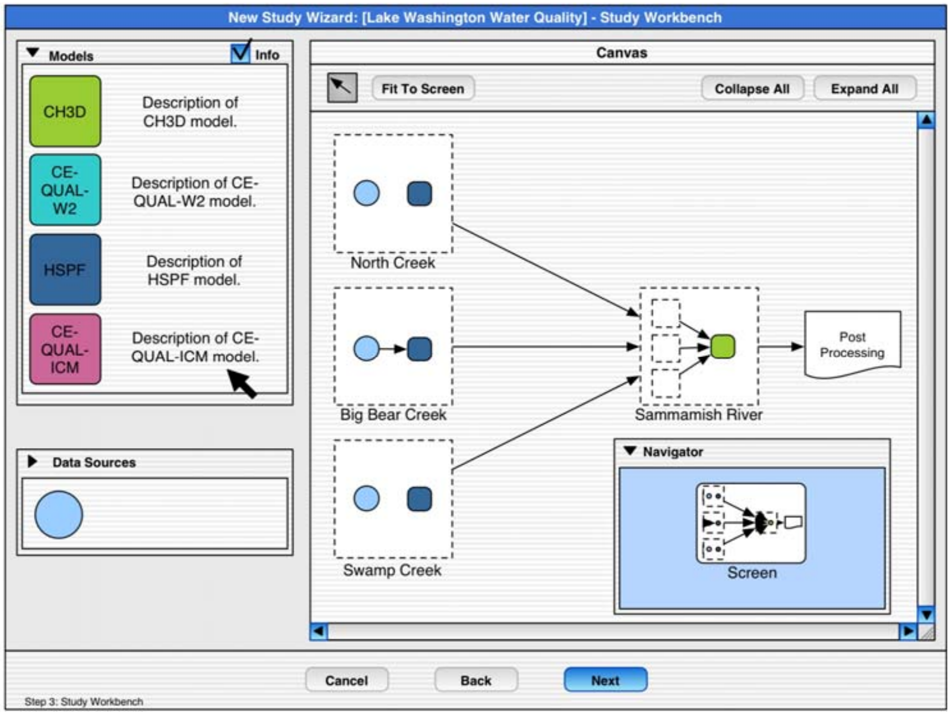
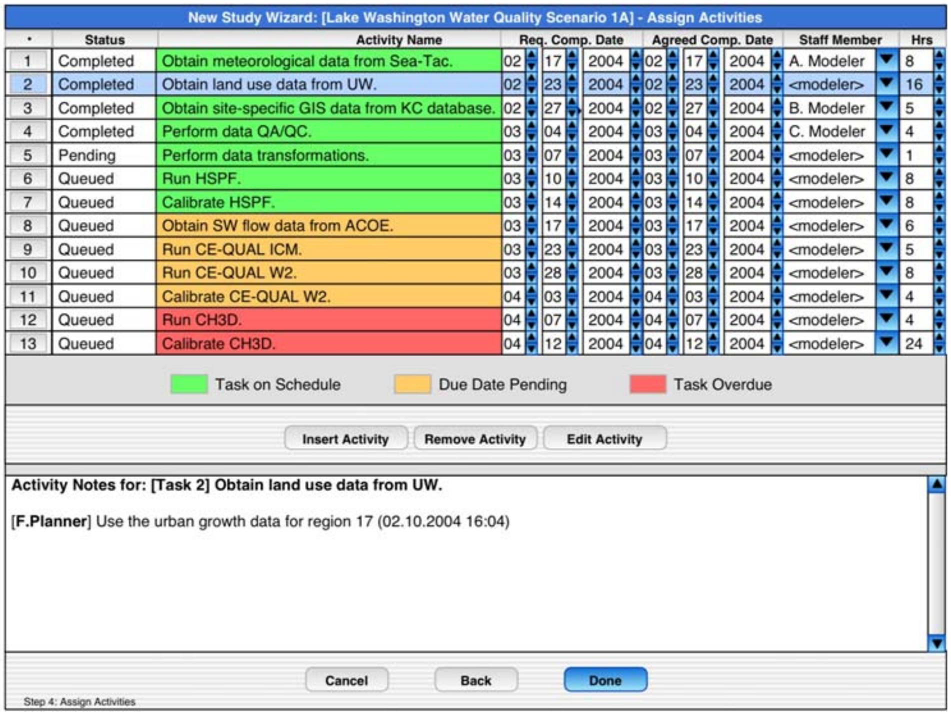


Figure 4 – Study Workbench

- Activity Assignment UI — At the management level, only the major components of the study are described. At the planner level, a deeper level of detail is required. A task decomposition stage translates the graphical elements from the study workbench into atomic activities. Activity assignment requires a link to King County's email system so to enable the dynamic assignment of tasks or collections of tasks to individual staff members (and specific computers as appropriate). For example, some staff might be tasked with specific data collection activities in support of a study, others with specific modeling and analysis activities, and other staff with follow-on risk assessment activities.



New Study Wizard: [Lake Washington Water Quality Scenario 1A] - Assign Activities

•	Status	Activity Name	Req. Comp. Date			Agreed Comp. Date			Staff Member	Hrs
1	Completed	Obtain meteorological data from Sea-Tac.	02	17	2004	02	17	2004	A. Modeler	8
2	Completed	Obtain land use data from UW.	02	23	2004	02	23	2004	<modeler>	16
3	Completed	Obtain site-specific GIS data from KC database.	02	27	2004	02	27	2004	B. Modeler	5
4	Completed	Perform data QA/QC.	03	04	2004	03	04	2004	C. Modeler	4
5	Pending	Perform data transformations.	03	07	2004	03	07	2004	<modeler>	1
6	Queued	Run HSPF.	03	10	2004	03	10	2004	<modeler>	8
7	Queued	Calibrate HSPF.	03	14	2004	03	14	2004	<modeler>	8
8	Queued	Obtain SW flow data from ACOE.	03	17	2004	03	17	2004	<modeler>	6
9	Queued	Run CE-QUAL ICM.	03	23	2004	03	23	2004	<modeler>	5
10	Queued	Run CE-QUAL W2.	03	28	2004	03	28	2004	<modeler>	8
11	Queued	Calibrate CE-QUAL W2.	04	03	2004	04	03	2004	<modeler>	4
12	Queued	Run CH3D.	04	07	2004	04	07	2004	<modeler>	4
13	Queued	Calibrate CH3D.	04	12	2004	04	12	2004	<modeler>	24

🟩 Task on Schedule 🟧 Due Date Pending 🟥 Task Overdue

Activity Notes for: [Task 2] Obtain land use data from UW.

[F.Planner] Use the urban growth data for region 17 (02.10.2004 16:04)

Step 4: Assign Activities

Figure 5 – Activity Assignment

- Connection to Other Features — This interface requires connections to other parts of the overall system, such as management of models and model revisions, browsing/querying of the data in the repository and data transform management.

Information and Actions Required from the County

A number of items need the County's attention:

- Verification of certain prior design considerations. Certain elements of the design, like how the client envisions using the study description field, were not adequately captured. Resolving these issues now will save time in redesign and implementation later.
- Active involvement as implementation progresses. Task 6 shall require feedback from County staff at the 6-, 12-, and 18-month milestones.
- Participation in formal usability studies to be performed at the Seattle office.
- Review of and comments on usability report.

Project Deliverables and Milestones

The main deliverables of this task are described below.

1. Finalized design sketches, across each of the main components, agreed upon internally (by Task Leads and implementation staff) and externally (client management and staff): February 20.
2. Functional prototype.

Milestone	**Deliverable**	**Completion Date**
Task Start		January 19, 2004
Paper Prototype Redesign	UI Design Document	February 20, 2004
Component Prototype		May 14, 2004
Combined Prototype	Internal Demonstration	May 28, 2002
Internal Usability Tests		July 2’ 2004
Redesign & Reimplementation	New UI Design Document	July 16, 2004
Integrated Demonstration	External Demonstration	Week of July 19, 2004
External Usability Tests		Week of July 19, 2004*

*External usability tests to follow the integrated demonstration.

Project Schedule

This current work plan is stacked heavily for the first six months in order to have a commanding presence at the demonstration in July. Work is scheduled to begin mid-January and continue uniformly until July. The subtasks listed above are not linearly dependent on one another, and some overlap is natural. The Task Lead shall review progress on each subtask and present it in monthly reports and during weekly Task-Lead meetings.

Key Staff

PNNL staff involved in Task 6 will include Andrew Cowell and Liam McGrath. The Task Leads (Randal Taira, Ranata Johnson, and Kevin Dorow) as well as PNNL IWRMS Project Manager Dave Thurman shall be the main conduits between this task and the others. Graphic design expertise will be added as necessary. Usability personnel shall be recruited as necessary to run usability tests as defined by the Task Lead. Documentation specialists shall be used as required.

Task 7 – Computing Architecture

Purpose

The purpose of Task 7, Computing Architecture, is to refine the IWRMS architecture to take advantage of dedicated computing hardware to be procured specifically for executing computational model runs, storage of input/output data, and maintenance of data pedigree metadata.

Key Objectives for Task 7

- Assist King County staff in identifying the optimum hardware and software for the dedicated computing cluster.
- Procure an initial "pilot" configuration of the computing cluster to support IWRMS development activities with the intent of transitioning the system to King County for further expansion and use as part of the IWRMS Version 1.0 deliverable.
- Integrate the functionality offered by the dedicated computing cluster into the IWRMS architecture.
- Test computing architecture functionality and develop documentation for it use by King County staff.

Scope

The Computing Architecture task will identify the requirements for and integrate a dedicated computing cluster into IWRMS. KC/DNRP requires that independent modeling tasks be executed in parallel on a dedicated computing cluster to reduce the calendar time required in completing studies. In addition, the computing cluster will provide dedicated resources for long running models (in excess of several days), thus alleviating the need for running models on the users' desktop machines and thereby reducing network traffic on the WLRD LAN. The functionality needed to support the Distributed Computing integration task will be documented and prioritized. This prioritization will be driven by the functionality relied upon by other components that are being concurrently developed within the IWRMS and by the Functional Requirements Specification.

Technical Approach

This task is a hardware / software acquisition, installation, and integration task combined with testing / documentation task for the resulting system. The software development task will be completed using the Java programming language and will leverage existing standards and technologies wherever possible. The steps required to complete the task have been broken down into following subtasks:

7.1 Determine Hardware for Dedicated Computing Cluster

Interface with King County staff and PNNL staff to determine the appropriate hardware to support a 10-15 node Windows-based computing cluster.

7.2 Determine Software for Managing the Dedicated Computing Cluster
Identify a COTS/GOTS software package that enables modeling jobs to be submitted from the IWRMS server for processing on the computing cluster. The package must run on Windows 2000 Professional (server and client) and support load balancing, job scheduling, and failure protection.

7.3 Install and Configure the Computing Cluster Hardware / Software for Development and Testing
Procure, install, and configure the identified hardware to support on-going IWRMS development tasks and integrate computing cluster functionality into the IWRMS system architecture. PNNL will install the cluster in their Seattle office to support development activities through at least the 18-month Demonstration.

7.4 Integrate the Dedicated Computing Cluster Functionality into the IWRMS
Develop a common software interface for other IWRMS components (Model Integration / Execution Wizard and Data Harvester) to take advantage of the computing cluster functionality.

7.5 Test the Dedicated Computing Cluster Functionality within the IWRMS

7.6 User Documentation
The IWRMS user documentation will be expanded to provide information on installing and operating the dedicated computing cluster as part of the IWRMS. The documentation will be provided as part of the IWRMS Documentation as defined in Task 10.

7.7 Task 7 Management
Task management will include tracking progress of subtask to make sure deliverables are met in terms of time and budget and providing monthly progress reports for the task.

Information and Actions Required from the County

County staff involvement will be required in the following tasks:

- Task 7.1—The County staff will be involved in determining the appropriate hardware for the dedicated computing cluster. Specifically, they need to identify any specific hardware requirements imposed by their existing IT infrastructure, any software/hardware requirements imposed by their intended use of IWRMS, and verify that they will be able to maintain the hardware solution selected.
- Task 7.6 – County staff may be involved in testing the functionality of the computing cluster as part of on-going testing expected to take place following the 18-Month System Demonstration.
- Task 7.7 – County staff will need to provide review and feedback of documentation associated with the computing cluster.

In addition, this task will require feedback from the County staff on an as-needed basis during the software development (which will take place via phone conversations / email when possible).

Project Deliverables and Milestones

Milestone	Deliverable	Completion Date
Determine Hardware / Software for Dedicated Computing Cluster (Task 7.1, 7.2)		November 22, 2004
Order Hardware/Software		November 30, 2004
Install and Configure the Computing Cluster Hardware / Software (Task 7.3)		January 17, 2005
Integrate the Computing Cluster into the IWRMS (Task 7.4)		February 14, 2005
Test the Dedicated Computing Cluster (Task 7.5)		February 28, 2005
Create the Dedicated Computing Cluster Documentation (Task 7.6)		June 21, 2005
	Dedicated Computing Cluster functionality to support 18-month demonstration	June 28, 2005
	Dedicated Computing Cluster to support IWRMS Version 1.0	November 2005

Project Schedule

Task 7 is estimated to take approximately six months to complete, with work starting in October 2004 and ending in March 2005. Limited task work will continue for the life of the project as necessary modifications are made and errors are addressed. The subtasks listed above are not linearly dependent on one another and some overlap is natural. The Task Lead will keep the Project Manager apprised of progress on each subtask in monthly reports.

Key Staff

PNNL/Battelle staff involved in Task 7 will include Dave Thurman, Ranata Johnson, Kevin Dorow, Liam McGrath, Alex Donaldson, Randal Taira, Karl Castleton, Justin Almquist, and Sharon Eaton.

Project/Development Management Approach

The following tasks (8, 9 and 10) outline the activities that will be used to mitigate project development and implementation risks.

PNNL will ensure that support for development activities is in place to ensure that project deliverables and milestones are met. This includes project management, requirements maintenance, configuration management, integration and system level testing, training and support for software delivered. Specifically, PNNL will:

- Apply project management controls to coordinate project activities, ensure that milestones are met, and ensure work is completed within budgets. In addition, the PNNL project manager or his delegate will attend meetings, provide periodic updates and briefings, manage the project risks that may arise when implementing tasks, and prepare monthly status reports. Monthly status reports will be submitted detailing the technical progress, including accomplishments and problems, cost status, and project risks. Periodic status briefings will be provided to the client immediately after each software release. These status reviews will be held at the client location.

- Ensure that the IWRMS Functional Requirements Specification, Revision 1 dated April 15, 2003 is updated and maintained to address the requirements specific to the development activities. This includes approval and prioritization of the requirements to ensure that the requirements are adequately documented and agreed upon by the KC/DNRP. All requirements will be prioritized by the KC/DNRP to ensure that the most important requirements are implemented early in the development life cycle. In addition, the requirements documentation will be maintained throughout the development life cycle to reflect any changes in scope as needed. All changes to requirements will require approval of KC/DNRP prior to implementation.

- Apply configuration management controls. This includes the process to build the software for internal testing, maintenance of the problem and issue tracking system, regular ongoing configuration management maintenance such as build support, and the release process.

- Verify proper operation of the IWRMS by conducting software and associated documentation testing prior to the installation of the IWRMS at the client location. The testing activities include the generation of an internal test plan that will describe how system and integration testing will be performed and who will conduct the testing. In addition the task includes the generation of tests, ongoing support of the test tools, execution of tests and test reporting throughout the development life cycle. The test reporting will include a summary of the test suite and the status of problems reported, fixed, closed, and any backlog identified by severity.

- Provide training support for interim prototypes and system builds as identified in the deliverables for each task. Training will include the generation of training materials and a workshop for selected King County users.

- Provide maintenance and support of the interim build of the components once they are delivered. This includes providing efficient off-site support, phone support to users in the field, and on-site support to the customer location as necessary.

- Pursue cost share opportunities with other parties that may have a need for similar IWRMS requirements. If successful in obtaining cost share opportunities, any cost share adjustments could be re-programmed within the project, with King County concurrence.

Proposed Project Schedule and Milestones

Figure 6 provides an estimated schedule of development activities described above. System design, development, and delivery are expected to take approximately 24 months, beginning 1 November 2003 and covering through October 2005. Two incremental prototypes are planned at six month intervals during the first year. The purpose of these prototypes is to demonstrate proposed IWRMS functionality and solicit King County feedback. Initial prototypes will be demonstrated at the PNNL Seattle office. The six-month prototype will focus on illustrating the basic functionality of the model integration subsystem, the data repository, and the study manager/UI components. The twelve-month prototype will feature enhancements to those components, plus additional capabilities offered by the model integration wizard, the external data harvester, and distributed computing capabilities (if adopted).

The initial delivery of the system is anticipated in the 18th month of the project. The initial system delivery (at the County) will give the client time to experiment with the system, identify bugs, missing features, or incompatibilities with the County computing environment. During the subsequent six months, as final system development activities are on-going, these issues will be addressed in priority order, subject to budget limitations.

The final system will be delivered during the 24th month of the project. At that time, the project will enter a long-term support phase during which system enhancements and bug fixes will be addressed through specific taskings from the County project manager, using a budget specifically allocated for those purposes.

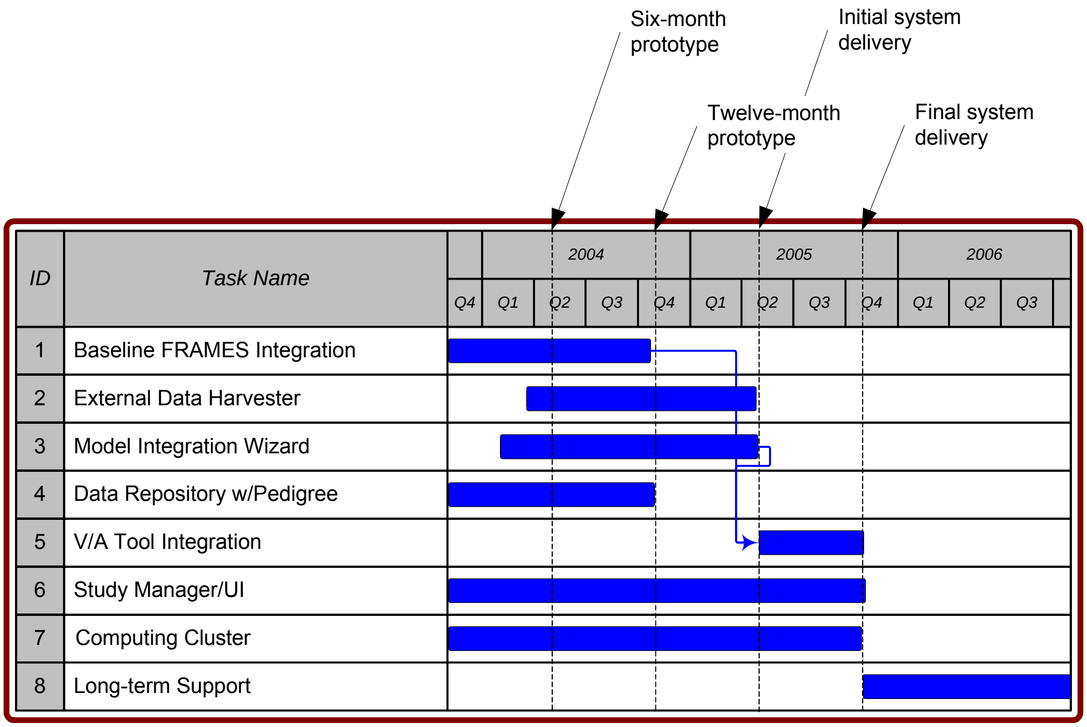


Figure 6 – Anticipated Project Task Schedule and Milestones

Task 8 – Software Maintenance and Support

Purpose

Once IWRMS Version 1.0 has been released and development is complete, the project will enter a maintenance and support. This task includes providing efficient off-site support, phone support to users in the field, on-site support to customer location, and providing maintenance release(s). As part of this task PNNL will:

- Maintain a database of software problem reports, suggestions, and enhancements based upon both on-site operation of the IWRMS software and development activities. PNNL will manage this database; determine corrective actions; and, with client direction and approval, issue maintenance versions of IWRMS to all users (up to quarterly if required).
- Provide IWRMS technical information to support IWRMS training and help desk support. Technical support will be provided directly to users and client as requested. Travel to the client site may be required to provide adequate support.
- Track costs associated with software maintenance and support separately from development activities. Scope adjustments of the IWRMS development task may be required to accommodate maintenance releases.
- Provide quarterly maintenance release(s) of the system as requested to address enhancements or fixes, as the budget and current scope of work permits. Major enhancements that affect the requirements may require the negotiation of additional scope of work.

Project Deliverables and Milestones

1. Maintain a problem reporting database.
2. Provide customer technical support as needed, including phone and on-site support.
3. Quarterly maintenance and enhancement releases as requested by King County.

Task 9 – Project Oversight

Purpose

The purpose of this task is to make sure that this project is a success, with the criteria for success being that: 1) PNNL meets its deliverables with respect to functionality, schedule, and budget and 2) the County is satisfied with the project progress and deliverables received.

Key Objectives

- Provide project oversight by tracking progress, deliverables, and budgets for all tasks.
- Verify that task leads are coordinating efforts so that interdependent system components will work properly when integrated.
- Be available to the County to address questions and concerns.
- Provide monthly status to the client.
- Oversee procurements and management of project resources.

Scope

Project tasks have been decomposed into technical components, and each of the tasks has an identified task manager. The task manager is required to produce a detailed work plan to make sure that that the project as a whole is successful. Overall project oversight and management is required.

This task includes the activities required to manage the overall project. These activities include client interaction, project planning and coordination, managing budgets, tracking deliverables, overseeing procurements, and reporting project status to the client. Each of these activities will be described in more detail in the following section.

Technical Approach

1. Client interaction
   The objective here is to maintain good communication with the County. Dave Thurman and Randal Taira in the Pacific Northwest National Laboratory (PNNL) Seattle office are available to County staff at any time to address project questions and concerns. Richland-based staff is also available on an as-needed basis to address detailed technical questions. We will also communicate PNNL staff questions and concerns to the County staff. Meetings with the County will also be coordinated through this task, whether they are requested by the County or by PNNL Task Leads.

2. Project Planning and Coordination
   This includes all of the overall planning and preparations involved in the project and will specifically include:

   - Working with the Task Leads to identify people with appropriate skills to add to the team.

- Planning a project kickoff meeting to present the project to the newly added project staff.
- Working with the PNNL financial and contract staff to manage project finances.
- Breaking down task budgets to specify design, testing, documentation, etc.
- Monitoring project progress and issues via weekly status meetings with Task Leads.
- Making sure that task leads coordinate their efforts as appropriate to enable us to meet overall project as well as task deliverables.

3. Managing Budgets
   Task Leads will be in charge of individual task budgets; however, as part of this task, we will monitor the budget in the bigger picture. Task Leads are required to assess task budgets on a weekly basis. The overall project budget will be reviewed and reported to the County Project Manager. Task progress and budget usage will be assessed and the client will be notified if issues develop.

4. Tracking Deliverables
   Task Leads will be in charge of task deliverables; however, as part of this task, further checks will be made to make sure that deliverables to the County are made. Of particular concern will be project deliverables that involve work from multiple tasks.

5. Overseeing Procurements
   The budget for all necessary project procurements will be managed out of this task, including hardware, software, and other materials deemed necessary for the successful completion of the project.

6. Reporting Project Status
   We will submit a monthly report to the County by the 10th of each month. This report will provide an overview of progress on each active task, a status report on the project budget, and a discussion of any changes in methodology, issues and problems, milestones made and pending, and budgets for their tasks.

Information and Actions Required from County

None

Project Deliverables and Milestones

Milestone	Deliverable	Completion Date
	Submit monthly report to the County	10th of each month
Coordinate with County for 6-month deliverable		July 2004
Coordinate with County for 12-month deliverable		December 2005
Coordinate with County for 18-month deliverable		June 2005
Coordinate system delivery		December 2005

Project Schedule

Task 9 will be active for the length of the project with work starting in December 2003 and ending in 2006 when long-term support ends.

Key Staff

Core staff involved in Task 9 will include Randal Taira and Dave Thurman, with other members of our multi-disciplinary staff involved as needed.

Task 10 – Implementation Support

Purpose

The purpose of Task 10, Implementation Support, is to make sure that appropriate support for Tasks 1 through 7 development activities is in place so that project deliverables are provided to King County (County) on time and within budget and that these deliverables are high quality. This task includes software task management, requirements management, configuration management, integration and system-level testing, training, and support for software delivered.

Key Objectives for Task 10

- Develop and implement good requirements management practices to make sure that all requirements are documented, tested, and accounted for throughout the development lifecycle.
- Develop and implement good configuration management practices to make sure that software and associated documentation changes are appropriately identified, managed, and controlled throughout the development lifecycle.
- Develop and implement integration and system-level testing activities throughout the lifecycle to verify that the products produced are adequately reviewed and tested before client delivery.
- Provide adequate training and support throughout the lifecycle to the client to enable the successful delivery of the software and associated documentation.

Scope

This task describes the engineering practices and methodologies to be applied throughout the product lifecycle of the Integrated Water Resource Modeling System (IWRMS). This includes using a standardized software engineering process to make sure that we deliver a quality product on time and within budget. This standard process will help identify and manage the risks inherent in computer system development by applying best practices in the areas of Project Management, Requirements Management, Configuration Management, and Reviews and Testing. This task also describes the training, support, and documentation needed to support the delivery of the IWRMS.

Technical Approach

10.1 Requirements Management

PNNL will update the IWRMS Functional Requirements Specification dated April 15, 2003 to address the requirements specific to the detail design and the implementation of the detailed task activities. This activity includes approval and prioritization of the requirements to make sure that the requirements are adequately documented and agreed upon by the County. All requirements will be prioritized by the County to make sure that the most important requirements are implemented early in the development lifecycle. In addition, the requirements documentation will be maintained throughout the development lifecycle to reflect any changes in scope as needed. All changes to requirements will require approval of the County before implementation. In addition, requirements will be managed and tracked so that no requirements are lost or any extra requirements added (without County and PNNL agreement) during development.

10.2 Configuration Management (CM)

PNNL will apply appropriate configuration management controls throughout the lifecycle. This includes developing and managing the process to build the software for internal testing, configuring the development environment, maintaining the problem-tracking system, and providing regular ongoing configuration management maintenance such as build support, and the release process. This task includes the generation of a project-specific Configuration Management Plan (CMP) that describes the following:

- Configuration management system used to control and maintain the software and associated data and documentation throughout the lifecycle.
- Requirements and resources necessary for the configuration management activities.
- Methodology for managing engineering changes during the product lifecycle.
- Software development environment and its associated directory structure, change control, and the software build process for all scheduled releases.

PNNL will make sure that the appropriate tools are in place to support configuration management activities. PNNL will use CVS for code and documentation control and PVCS Tracker for problem and issue tracking.

10.3 Integration and System Testing

PNNL will verify proper operation of the IWRMS by conducting software and associated documentation testing before installing IWRMS at the client location. The testing activities include the generation of an internal test plan that will describe how integration and system testing will be performed and who will conduct the testing. The task also includes the generation of tests (automated whenever possible), ongoing support of the test tools, execution of tests, and test reporting throughout the development lifecycle. The test reporting will include a summary of the test suite and the status of problems reported, fixed, closed, and any backlog identified by severity. The test report will be delivered to the client after the final 18-month delivery.

PNNL will apply an iterative testing approach that addresses both integration and system-level testing. Integration testing is the orderly progression of testing in which software elements, hardware elements, and operator interfaces are combined and tested, until all inter-module communication links have been integrated. Integration testing focuses on the interfaces between modules. PNNL intends to schedule iterative builds throughout the development cycle that integrate different pieces of the system so that they can be adequately tested before the entire system is ready. This enables problems to be discovered as early in the development lifecycle as possible so they can be fixed before delivery to the client.

System testing involves executing the completely integrated system to validate that the software meets its requirements. PNNL will verify conformance to functional and performance requirements through the generation of a Test Plan. During development of the Test Plan, each system requirement will be reviewed and may be restated in terms of attributes that can be measured during the testing process. In addition, the documentation products that will be delivered will be adequately reviewed and tested to verify that the documentation reflects the delivered system. PNNL will involve the County in system-level testing activities as the system progresses.

10.4 System Administration

PNNL will make sure that the IWRMS is managed and maintained throughout the development lifecycle. PNNL will appoint a System Administrator to manage the overall software development environment, including regularly and systematically scheduled backups. The System Administrator will support various aspects of configuration management and testing activities and will provide support for project lab development, training, and production systems (hardware and software).

10.5 Documentation

PNNL will generate the documentation necessary to support the scheduled releases of the components for Tasks 1 through 7. Each scheduled release will be accompanied by the associated documentation needed. PNNL expects to generate the following documentation:

- A User's Guide that provides the details necessary for the client to successfully use the system, including detailed descriptions of the mapping of the different data types, aid in using the Model Integration Wizard, aid in using the data repository functionality, instructions to support the use of the Study Manager and its various components, and any updates needed to support a distributed computing environment.
- An Installation Guide that provides the instructions for installing the system including the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES) integrated baseline and all scheduled releases.
- An Administration Guide that provides the information necessary to successfully administer the system, including administration of the data repository and a distributed computing environment (if implemented).

10.6 Training and Support

PNNL will provide training support for interim builds of the components for each task. Training will include the generation of training material, a workshop for selected users, and travel to the client's office.

PNNL will provide maintenance and support of the interim builds of the components once they are delivered. This includes providing efficient offsite support, phone support to users in the field, and onsite support to the customer location as necessary.

10.7 Task 10 Management

Task management will include tracking progress of subtasks to make sure that deliverables are met. It also includes assessing progress to internal milestones to make sure that issues related to scope, schedule, and budget are addressed in a timely manner.

This task also will be responsible for the overall software management process to enable successful integration among Tasks 1 through 7. This includes development of an integrated project schedule to make sure that builds, testing, and documentation are successfully integrated with development activities. This task will provide monthly progress reports by the 5th of every month.

Information and Actions Required from the County

- Review and approval of prioritization of requirements and timely review of documented requirements for completeness before each scheduled release or interim build.
- Review of interim software deliverables/builds to make sure that client issues can be resolved in a timely manner. The County's active participation in testing the software once delivered at the interim builds is strongly encouraged. Documenting concerns with design implementation early in the development cycle is easier to fix and less costly to the project. Timely review and approval of these products as they are available will be appreciated and will enable us to remain on schedule and within budget for this task.

Project Deliverables and Milestones

Milestone	Deliverable	Completion Date
Microsoft Project schedule (6-month demonstration)		March 19, 2004
Configuration management plan complete		April 30, 2004
Functional requirements specification updated for 6-month demonstration	Draft functional requirements specification for County review	June 11, 2004
	[County Deliverable to PNNL] County review of updated functional requirements specification complete	June 18, 2004
Update FRAMES test suite to reflect changes		June 30, 2004
Microsoft Project schedule (12-month demonstration)		October 1, 2004
Functional requirements specification updated for 12-month demonstration	Draft functional requirements specification for County review	December 9, 2004

Milestone	**Deliverable**	**Completion Date**
	[County Deliverable to PNNL] County review of updated functional requirements specification complete	December 16, 2004
IWRMS test plan complete (12-month demonstration) October 2004		December 17, 2004
Build and test (12-month demonstration)		January 10, 2005
IWRMS test report complete (12-month system)		February 11, 2005
Microsoft Project schedule (18-month deliverable)		March 4, 2005
IWRMS test plan complete (18-month deliverable)		April 28, 2005
Functional requirements specification updated for 18-month deliverable	Draft functional requirements specification for County review	May 31, 2005
	[County Deliverable to PNNL] County review of updated functional requirements specification complete	June 7, 2005
Build and testing (18-month deliverable)		June 2005
IWRMS test report complete (18-month deliverable)		July 2005
IWRMS Beta Version software system complete	IWRMS Beta Version software system	October 2005
IWRMS draft documentation complete	Functional requirements specification, design documents, user's guide, administration guide, and installation guide as necessary for each of Tasks 1 through 7. (All draft versions)	October 2005
Training complete		October 2005

Milestone	Deliverable	Completion Date
Build and testing (IWRMS Version 1.0)		November 2005
IWRMS Version 1.0 software system complete	IWRMS Version 1.0 software system	December 2005
IWRMS Version 1.0 documentation complete	Functional requirements specification, design documents, user's guide, administration guide, and installation guide as necessary for each of Tasks 1 through 7	December 2005
IWRMS test report complete (IWRMS Version 1.0)	IWRMS test report	January 2006

*Build and test schedule cycle will be determined as soon as more detailed Microsoft Project schedules are developed.

Project Schedule

Task 10 consists of ongoing activities throughout the lifecycle and is estimated to take approximately eighteen months to complete, with work starting in February 2004 and ending in December 2005. The Task Manager will keep the Project Manager apprised of our progress on each subtask in monthly reports.

Key Staff

PNNL staff involved in Task 10 will likely include Dave Thurman, Ranata Johnson, Kevin Dorow, Randal Taira, Brian Homer, Julie Dunkle, Nancy Holter, Sharon Eaton, and Eric Robinson.

Appendix A

Configuration Management Plan For the Integrated Water Resource Modeling System (IWRMS) Project

Table of Contents for Appendix A

1.0 Purpose

The purpose of this Configuration Management Plan is to describe the configuration management system used to control and maintain the IWRMS software, associated data, and documentation throughout the life cycle of the project. It details the requirements and resources necessary for Configuration Management (CM) activities within the IWRMS project. It establishes the methodology for managing engineering changes during the product life cycle. It describes the software development environment and its associated directory structure, change control, and the software build process for all scheduled releases.

2.0 Scope

This plan defines the configuration management related activities that will be implemented for all software development products and processes throughout the applicable life cycles of the project. This plan is effective May 1, 2004.

3.0 CM Organization and Personnel

Overall responsibility for configuration management resides with the Implementation Support Task Lead, who may designate an appointed representative, referred to within this document as the Configuration Manager. The Implementation Support Task Lead will ensure that an effective CM program is established, budgeted, implemented, and maintained. The Implementation Support Task Lead or the designated Configuration Manager is responsible for establishing procedures to implement CM activities. The specific CM responsibilities of various team members are listed in the next section.

The project team consists of managers, developers, testers, and documenters responsible for the development, enhancement, and integration of their respective design functions or software applications, together with support teams that provide other project services. All team members are accountable for implementing the required CM procedures and practices in their respective areas of responsibility. Team members will often have more than one role.

CM Roles and Responsibilities

This section defines the CM roles and responsibilities of all project team members. Because CM is a project-wide function that relates to most tasks, it is of concern to all team members. In addition, key CM roles and responsibilities are assigned as described below:

Implementation Support Task Lead

In the area of CM, the Implementation Support Task Lead's responsibilities are to:

1. Ensure that a CM Plan is prepared as part of the project's engineering practices and that it reflects the project requirements.
2. Identify a Configuration Manager to implement CM.
3. Provide funding, scheduling, personnel, and other resources for CM.

4. Consider and process all change requests for software, data, and documentation. This includes verifying the corrective action, if required, assigning the action, and confirming its satisfactory completion. May delegate responsibilities as needed.
5. Ensure that releases are appropriately identified and maintained per the mechanisms outlined in this plan for software, data and documentation.
6. Ensure that releases for software, data and documentation are distributed in accordance with this plan.
7. Ensure appropriate CM controls are applied to software releases, upgrades, patches, and installation of fielded systems.
8. Establish and document appropriate CM procedures to implement CM activities related to operations. This includes automation of installation scripts whenever possible.

Configuration Manager

The Configuration Manager works in close conjunction with all task leads to maintain the integrity and traceability of software, data, and documentation changes; and to build version-controlled product releases. Specifically, the Configuration Manager shall:

1. Assist in the preparation of the CM Plan, and take responsibility for its implementation.
2. Establish and maintain a mechanism to identify and control software, data, and documentation (this includes version control).
3. Use the PVCS Tracker automated tracking system, process, and track change requests.
4. Establish and maintain the PVCS Tracker to show the status of all changes as they are proposed, approved, and implemented.
5. Establish file naming and numbering conventions, as required.
6. Work with the Build Manager to support build activities, as necessary.
7. Generate status accounting reports and other CM documentation, as required.
8. Ensure that project team members are adequately trained in CM procedures and in the use of CM tools.
9. Support quality assessment and improvement of the CM process by giving appropriate feedback and recommendations.
10. Work with the project team to establish appropriate CM procedures to implement the CM activities.
11. Maintain agenda and minutes for Configuration Control Board (CCB) meetings.

Project Manager (PM)

The PM has the following CM-related responsibilities:

1. Monitor CM status reports prepared by the Configuration Manager, review other relevant reports, and take appropriate action to facilitate and improve CM.
2. Act as the final authority on the disposition of all change requests submitted.
3. Acts as the Chair of the CCB.

Task Leads

The individual task leads are responsible for ensuring that project team members associated with their tasks follow this CM Plan. Specifically, the Task Lead's CM responsibilities include:

1. Ensuring that project team members read, understand, and follow the practices and procedures identified in this CM Plan.
2. Bringing CM issues to the immediate attention of the Configuration Manager or Implementation Task Lead.
3. Serving on the CCB.
4. Identifying issues that need CCB attention.
5. Supporting the Implementation Task Lead on CM issues when needed.

System Administrator

The System Administrator manages the overall software development environment. In terms of CM, the System Administrator is expected to:

1. Establish user accounts and access permissions.
2. Install suitable version control software for the project, other tools as necessary.
3. Ensure controlled access to all project files, including code, documentation, development tools, and utilities.
4. Ensure that files are regularly and systematically backed up.
5. Establish database accounts and access permissions as requested.
6. Upgrade development, test, and production databases as requested.

Build Manager

The Build Manager assembles successive builds for testing or for product and data release and delivery to the client. The Build Manager will:

1. Develop the necessary build scripts to create a successful build.
2. Document the build process as a procedure.
3. Perform builds.
4. Generate necessary release notes for each product and data releases.
5. Support product release preparation and distribution (documentation and media).

Project Team Members

CM affects all members of the project team in different ways depending on their specific tasks and assignments. As a minimum, each project team member has an individual responsibility to:

1. Know and understand what the CM requires concerning his or her assigned tasks, and ensure compliance with those requirements (this includes version control).
2. Acquire a reasonable proficiency in the use of CM tools. Including regularly checking of their Tracker In-Tray.
3. Maintain the integrity of the project software, data, and documentation by following proper CM procedures in all task execution.
4. Participate in improving the CM process by providing appropriate feedback and recommendations to the Configuration Manager or Implementation Support Task Lead.

Configuration Control Board (CCB)

The CCB has been established to ensure that all Software Change Requests (SCRs) generated from the IWRMS project and FRAMES project using shared code are reviewed for successful reuse and sharing between the two projects. The CCB includes the IWRMS Project Manager, who acts as the chair at its meetings, the IWRMS Configuration Manager, who maintains the agendas and minutes of the CCB meetings, and other IWRMS Task Leads as required. The Project Manager from the FRAMES development project will serve on the CCB and may designate additional CCB members as needed. As a group, the CCB has the following responsibilities:

- Consider and process all SCRs for software and documentation that are brought to the CCB for disposition. This includes identifying the corrective action, if required, assigning the action, and confirming its satisfactory completion.
- Monitor CM status reports prepared by the IWRMS Configuration Manager, reviewing other relevant reports, and taking appropriate action to facilitate and improve the CM process.

Staff Roles

Table 1 below shows the initial roles of each staff member as it relates to their CM roles and responsibilities:

CM Role(s)	Team Member	Member of CCB
Implementation Support Task Lead, Configuration Manager	Ranata Johnson	Yes
Project Manager	Dave Thurman	Yes
Task Lead, Project Team Member, Build Manager	Kevin Dorow	Yes
Task Lead, Project Team Member	Randal Taira	Yes
Task Lead, Project Team Member	Andrew Cowell	Yes
Project Team Member, System Administrator	Brian Homer	
Project Team Member, Back Up Build Manager (Seattle Office)	Liam McGrath	
Project Team Member	Eric Robinson	
Project Team Member	Don Flynn	
Project Team Member	Sharon Eaton	
Project Team Member	Mitch Pelton	
Project Team Member	Karl Castleton	
Project Team Member	Bonnie Hoops	
FRAMES Development Project Manager	Gene Whelan	Yes
Project Team Member	Linda Deatherage	

Table A1 – Initial CM Role/Responsibility Matrix

4.0 Tools

CM Tools

Software tools are used to facilitate CM by tracking the identification, control, and status accounting of system software, data, and documentation. These tools allow for management of software revision and release control in a multi-developer environment. The CM tools designated for use are:

- Concurrent Versioning System (CVS) for source code control and documentation control. The project team will use Tortoise CVS, which lets you work with files under CVS version control directly from Windows Explorer.
- Apache Ant (Version 1.6.1) will be used to produce the IWRMS builds. This is a Java-based build tool and is open source.
- PVCS Tracker is used to manage the problem reporting process described in Section 6.0, Software Change Request (SCR) Process.
- Other general-purpose tools, such as SharePoint and Microsoft Office tools may be used to generate administrative and status reports, as required.

Development Environment Tools

A consistent development environment has been created for the project. The development environment and tools that will be used for the project are:

- Together Control Center software is used to document the design of the IWRMS functionality and support the automatic generation of J-Unit test suites. Design tools are used to facilitate the design process and provide automated documentation whenever possible.
- Java Version 1.4.2_03 will be used for development of new functionality.
- SQL Server 2000 Service Pack 3 will be used for the database and repository development.
- FRAMES Version 2.0 is the base for the shared code to be used for IWRMS.

5.0 Configuration Identification

This section describes the overall structure of the IWRMS development environment, the concept of configuration items, and the process of putting configuration items under configuration control. This includes source code modules and packages in Java, scripts, database management files, and the design and user documents associated with IWRMS software.

Project Directory Structure

A standard hierarchical directory structure will be established to facilitate the development, reuse, and maintenance of the IWRMS technology, electronic documentation, and software source code. Figure A1 shows the structure identified for the project. Note that IWRMS will use/adopt the current structure of FRAMES 2.0 for the shared code.

File Locations

The projects items are stored in various locations for the project as noted below:

- Source code modules, scripts and database management files are located on \\KingCountyDev\cvsroot
- Project, design and user documents are located on the project's SharePoint server at http://pnnlsp1/sites/Kingco

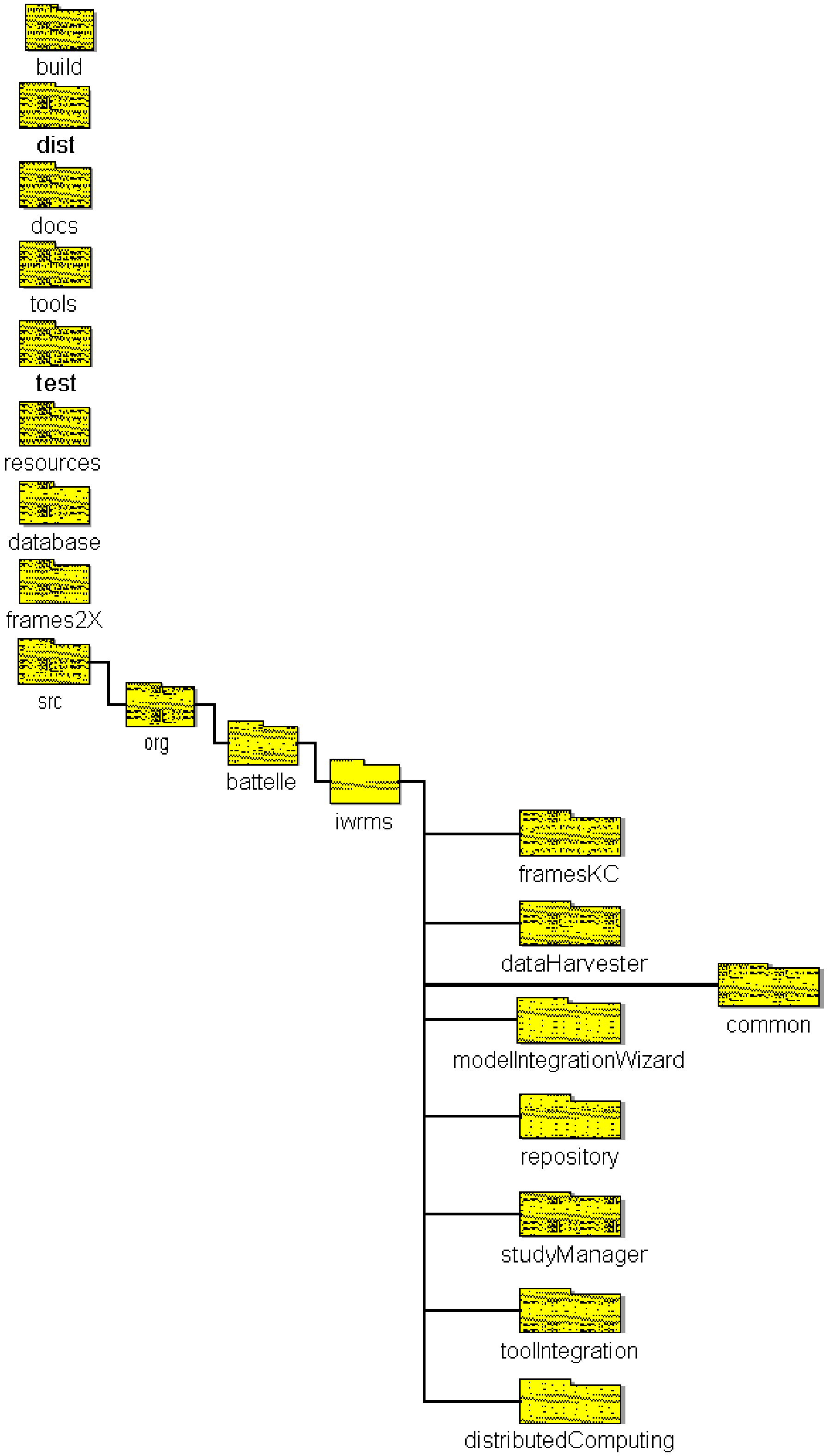


Figure A1 – IWRMS Directory Structure

Promotion Model

The project shall use the following promotion model for all project Configuration Items (CI) as they progress through the life cycle. A configuration item is defined as an aggregation of hardware, software, or documentation that is designated for CM and treated as a single entity in the configuration management process. For example, the Data Harvester portion of IWRMS is considered a configuration item. The promotion model consists of four states as shown in Figure A2:

- Development
- Testing
- Demonstration
- Production

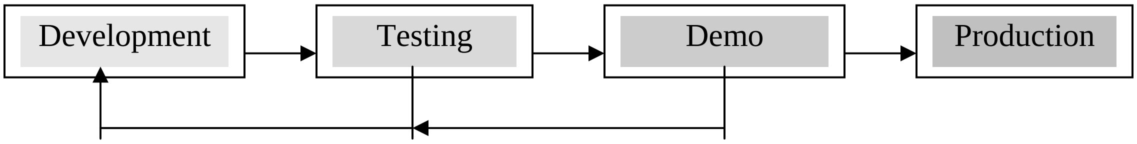


Figure A2 – IWRMS Promotion Model

It is not acceptable for a CI to skip a state in the model. At each state, the operations allowed on a CI will differ as follows:

- In Development, informal change control is applied to each CI, and all project members may access all CIs. At all other states, formal configuration control is applied.
- In the Testing state, only the testers may access the CIs.
- In the Demo state, the complete package is evaluated and used for demonstrations before its official release.
- In the Production state, the product is ready for official release. The backward arrows indicate that a CI may be demoted if it fails to pass a test or satisfy a requirement.

The Commercial Off-The-Shelf (COTS) software that is unmodified and is linked into the IWRMS software package (e.g., ArcView, MATLAB) shall retain its original identification by name and version (i.e., it shall not be re-identified as an IWRMS CI). Documentation specific to the COTS software shall be limited to the existing commercial documents.

6.0 Software Change Request Process

An internal configuration status accounting system has been established to track SCRs. The status of proposed changes is progressively tracked by the Configuration Manager through approval and implementation by the Implementation Support Task Lead. These records are used to provide progress reports from the Configuration Manager to the Project Manager and other Task Leads and to guarantee traceability between releases.

Software Change Request (SCR) Process

This section describes the SCR process, by which software items are modified to fix problems or add new features to the software or documentation. All modifications are controlled via the SCR process. Once an SCR is logged into the PVCS Tracker system, it remains open until:

1. All changes and tasks associated with the SCR have been implemented and verified or
2. The SCR is disapproved.

A SCR may be submitted by any individual on the project. Project team members will enter all SCRs online, using PVCS Tracker. No paper copy is necessary. The Implementation Support Task Lead or delegate will designate a specific individual to own a SCR once an initial evaluation of the SCR is complete. All SCRs undergo this initial review. This evaluation analyzes the problem, verifies the severity, determines if the requested modification is within the scope of present work, determines the impact to the shared code base, and proposes solutions. The Implementation Support Task Lead may also disposition the SCR so no changes are implemented at this time but the SCR remains open by marking the resolution in PVCS Tracker to “Do Not Fix”, or “Deferred”. Periodically, the PM may be consulted on disposition of SCRs. Any SCRs that change requirements must be approved by the IWRMS Project Manager before implementation. In addition, periodically, members of the CCB may be asked to process SCRs to decide their disposition. Any SCRs affecting the shared code base with FRAMES must be approved by the CCB before implementation.

When an SCR is approved for implementation, the SCR is assigned to a member of the team responsible for implementing or fixing the item. The revised item is then tested. Once it passes testing, the Implementation Support Task Lead verifies the test results and closes the SCR. The Figure A3 details the SCR Process.

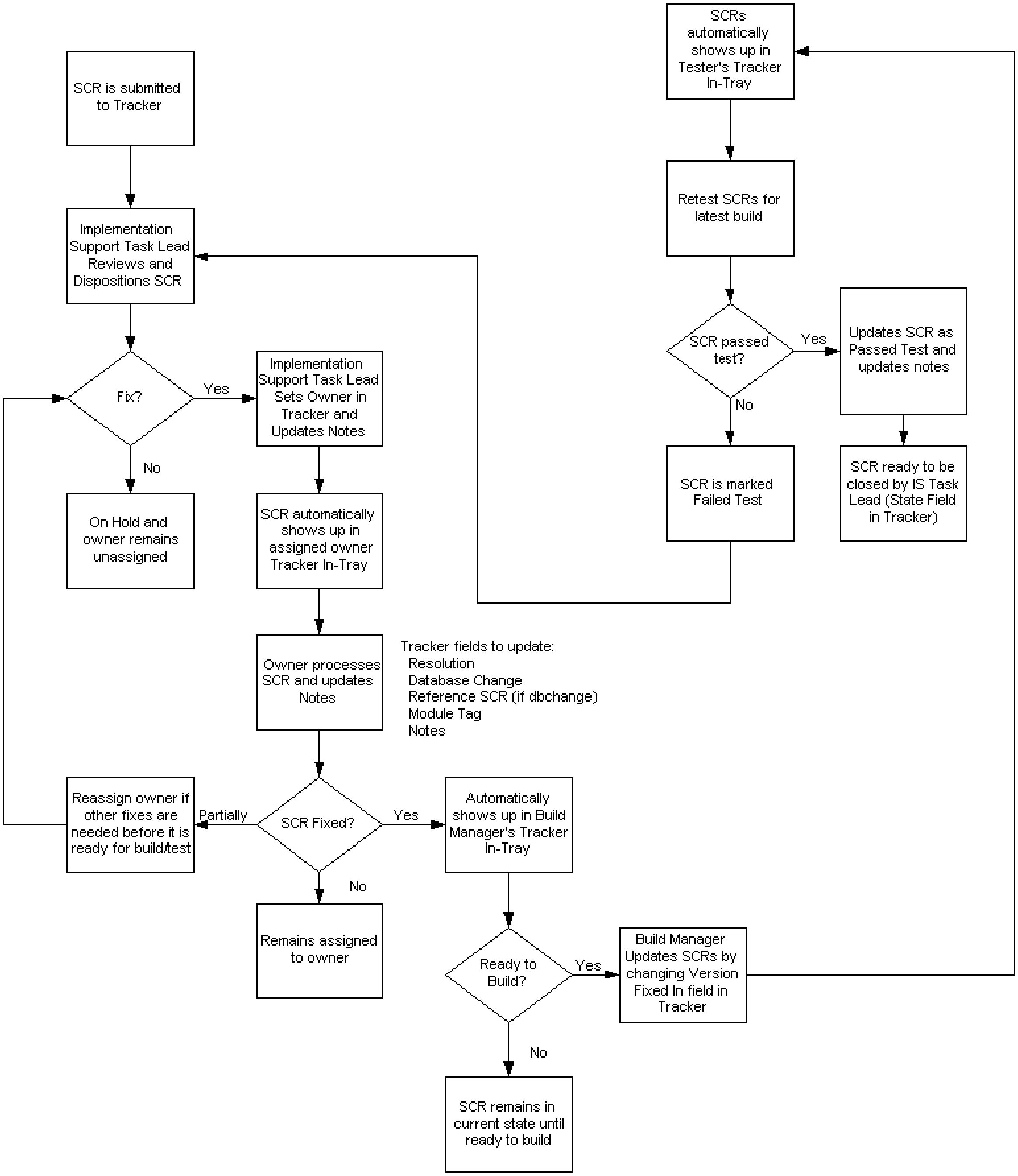

SCR is submitted to Tracker
Implementation Support Task Lead Reviews and Dispositions SCR
Fix?
Yes
No
Implementation Support Task Lead Sets Owner in Tracker and Updates Notes
On Hold and owner remains unassigned
SCR automatically shows up in assigned owner Tracker In-Tray
Owner processes SCR and updates Notes
Tracker fields to update:
Resolution
Database Change
Reference SCR (if dbchange)
Module Tag
Notes
SCR Fixed?
Partially
Yes
No
Reassign owner if other fixes are needed before it is ready for build/test
Remains assigned to owner
Automatically shows up in Build Manager's Tracker In-Tray
Ready to Build?
Yes
No
Build Manager Updates SCRs by changing Version Fixed In field in Tracker
SCR remains in current state until ready to build
SCRs automatically shows up in Tester's Tracker In-Tray
Retest SCRs for latest build
SCR passed test?
Yes
No
Updates SCR as Passed Test and updates notes
SCR ready to be closed by IS Task Lead (State Field in Tracker)
SCR is marked Failed Test

Figure A3 – IWRMS SCR Process

7.0 Code Management

Throughout the life cycle of a software product, the functional requirements and component designs evolve in a controlled fashion for each scheduled release. Code management consists of procedures for: 1) placing software items under CM, 2) processing SCRs, and 3) creating new versions/revisions of software and associated documents. Figure A4 shows the IWRMS development environment used to manage this process. This section outlines those ongoing CM activities that occur during the life cycle of code development.

Code Identification

Configuration of the code consists of defining each item of software, data, and documentation under development by the project, naming or labeling these items, and accurately capturing and preserving their various revisions or versions. The FRAMES shared code will retain the original identification scheme currently in use by the FRAMES project team.

The Configuration Manager shall define, and the project team will implement, a method of identifying software source code to ensure adequate traceability. The identifying information will be included in a header within each file, if possible. Table 2 below shows the standard component naming convention that shall be applied:

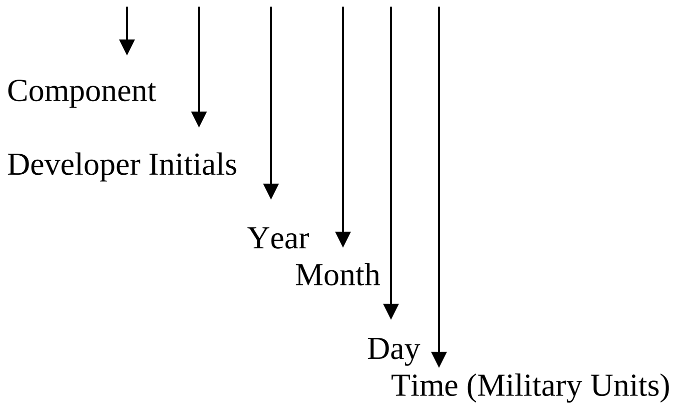
Component Name	Naming Convention
Model Connection Framework (FRAMES) King County	MCF
Data Harvester	DH
Model Integration Wizard	MW
Repository	RP
Study Manager	SM
Tool Integration	TI
Distributed Computing	DC
Common	CM

Table A2 – Standard Component Naming Convention

Moreover, the CVS Tag will be applied in code as changes are checked in. The format for the CVS Tag will be:

componentnamingconvention_developerinitials_date/time stamp

Example: DH_RLJ_ 2004-03-19-0911



The Configuration Manager shall ensure that all source code, configuration files, scripts, and data residing on the project development platform are identified through the use of CVS.

For the IWRMS project, a major release number, minor version number, and a build number will apply to all software builds (for testing or product release). This number shall have the following content:

Major Version	n-digit release number to be assigned by the Implementation Support Task Lead (initial value = 0)
Minor Version	n-digit version number to be assigned by the Implementation Support Task Lead (initial value = 1)
Build Number	nnn-digit version number to be assigned by the Build Manager (initial value = 001).

Example: IWRMS Version 1-1-001

Code Control

Configuration control consists of procedures for: 1) placing software items under CM, 2) processing SCRs, and 3) creating new versions/revisions of software and associated documents. The flow of configuration control is described in the following paragraphs and shown in Figure 4.

After completing the startup work on an initial version of a source code file, the developer initiates CM by promoting the file to the ***shared archives***. This is done by checking in an initial version of the file to the CVS. The developer will do this after unit tests are conducted on the module and prior to integration testing. In any case, the CVS check-in procedure will ensure that information about the file from the developer, e.g., CVS Tag, unit test status, dates, and revision numbers, is retained in a header of the source code file, which accumulates tracking data for the archive file throughout the life of the project. From this point on, the file is under CM control, and all changes made to this file must be documented on an SCR form.

When the SCR is assigned to a developer to fix the code or add a new feature, the developer checks out the most recent archived version of the file. As program development progresses, the developer maintains a working file in a private (not shared) directory. The developer may choose to work along a branch revision tree. This preserves the original revision as the current archived source for other developers to work from, while branch revisions are made in the developer's private files.

When the code goes through a build cycle, each developer ensures that the appropriate code is checked in to the working shared environment. The developer will provide via email the modules and their associated module tags ready for build to the Build Manager.

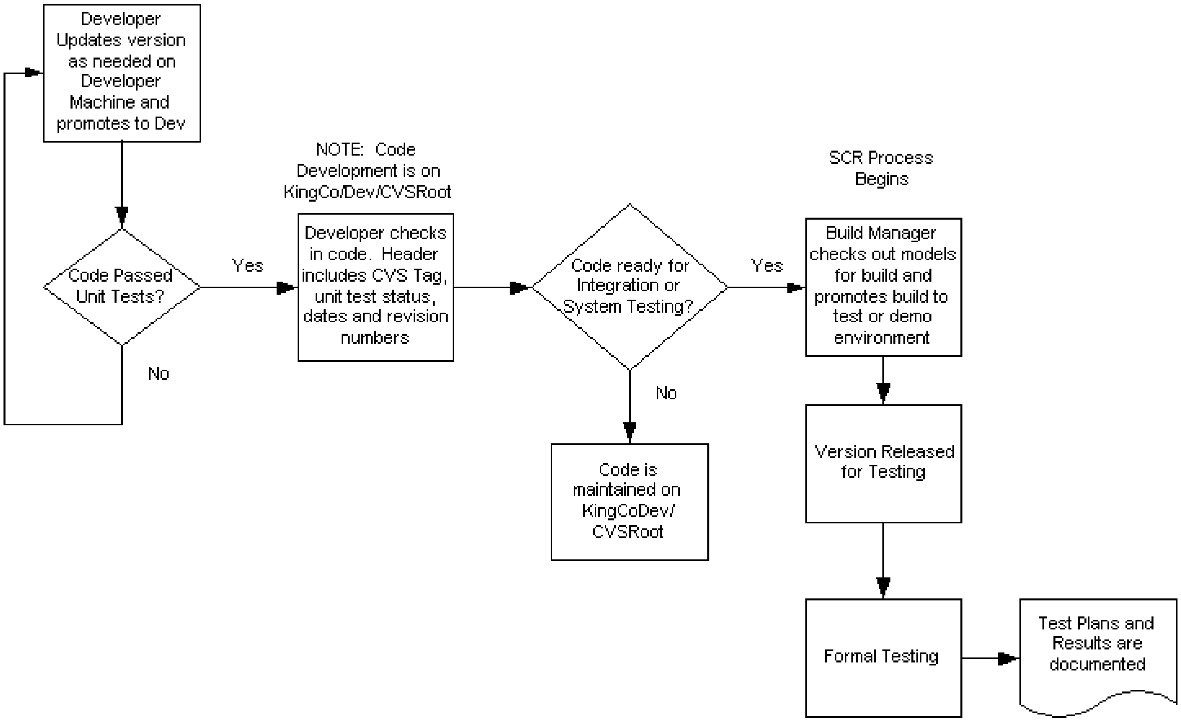

Developer Updates version as needed on Developer Machine and promotes to Dev
Code Passed Unit Tests?
Yes
No
NOTE: Code Development is on KingCo/Dev/CVSRoot
Developer checks in code. Header includes CVS Tag, unit test status, dates and revision numbers
Code ready for Integration or System Testing?
Yes
No
Code is maintained on KingCoDev/ CVSRoot
SCR Process Begins
Build Manager checks out models for build and promotes build to test or demo environment
Version Released for Testing
Formal Testing
Test Plans and Results are documented

Figure A4 – IWRMS Code Management Process

8.0 Software Release Process

This section describes the software release process. The software release process is an iterative process that is completed for informal builds (for purposes of testing, etc.) or formal product releases throughout the lifecycle. The release process is more formalized when the project gets to the final code freeze. This software release is a multi-step process beginning with a code freeze and leading to the build, system testing, media production, document collection, and packaging, installation, and patching. IWRMS software will be released per the schedule in the Microsoft Project Schedule.

Code Freeze

Before a software release/build occurs, the Implementation Support Task Lead declares a code freeze schedule for the specific build. This signals to the developers when they must complete their coding, assess their efforts, and move any completed and unit tested modules to the source code repository. This is the first step in the preparation for the build. A code freeze for the project is defined as a point in time for any given release after which no changes can be incorporated into the specific build.

Software Build

The Build Manager develops an appropriate list of all elements of the software designated for release. Each release will require a software build that includes source code, drawn from the central archive, is compiled into a suite of executables. All software releases will be properly identified as noted in previous sections of this plan. The detailed procedures for software builds using Apache Ant will be defined in the IWRMS Builds Documentation, which is not yet available.

Product Testing

The product suite is staged by performing an installation and functional test. This ensures that the system complies with the planned release criteria and that the product to be delivered contains all the required files for successful installation and operation at the customer site.

Media Production

The IWRMS software is distributed on a ***CD or DVD as necessary***. These media are verified for content by inspection and comparison of the contents. The initial delivery to the client will be on the server maintained at the PNNL Seattle Office.

Document Collection

The IWRMS software releases will be accompanied by a set of documentation consisting of the *User's Guide*, *Administration Guide*, and *Installation Guide*. These documents are considered part of the software releases and will be delivered electronically (format not yet determined). These documents are delivered with the IWRMS software.

Packaging and Installation

The product suite, consisting of the software and documentation, is labeled with appropriate copyright and disclaimer information along with product and version identification. The IWRMS product is packaged and installed for the client via the network or travel to the Seattle location of the installation.

Patching the Software

It is always the intent of the project to deliver the highest quality software that is possible. It is not the philosophy of this project to patch the system, but it may become necessary to ensure that the users have a usable operational system. The definition of a patch is a necessary fix because the Project Manager has deemed the software problems to have serious impacts to the operability of the overall system and may at times be embarrassing.

When patches are created the client will be notified and the patches will be delivered to the client or made available to the client via a website.

The Configuration Manager will work with the Project Manager to ensure that the approach for the patch is well defined and that the guidelines for CM are followed. Approved changes for a patch will follow the same process outlined for product release.

9.0 CM Plan Training

All project staff are required to read the CM Plan. Project staff will be required to read other configuration management documents as directed by the Implementation Support Task Lead.

10.0 CM Plan Maintenance

Maintenance of this plan is necessary to ensure continued CM planning during the release life cycle of the product. It is the responsibility of the Implementation Support Task Lead to monitor compliance with this plan and ensure that changes and updates to the plan are implemented, as required. At a minimum, reviews of this plan will occur at the start of each new product release. At that time, proposed changes, if any, will be evaluated and, with the approval of the Project Manager, will be implemented within the plan. All plan changes will be communicated to the project team.

11.0 Revision History

Release	Date	Comment
0.1	February 19, 2004	Initial draft to task leads for comment
0.2	March 23, 2004	Second draft out for comment
0.3	April 17, 2004	Last round of comments incorporated, sent to IWRMS Project Team for review
0.4	April 26, 2004	Comments incorporated from Sharon Eaton, Kevin Dorow, Randal Taira, Eric Robinson, Liam McGrath, Gariann Gelston. Sent to Dave for final review. **NOTE: Staff received training against this version.**
0.5	August 25, 2004	Incorporated Dave's comments per his review on 7/13/04 (minor changes only)
0.6	November 2, 2004	Put document through ERICA for external distribution - PNNL-14929 Rev 0

IWRMS and FRAMES Future Upgrades and Enhancements

Introduction

This document describes plans for continued, simultaneous development of IWRMS and FRAMES and conditions for future system upgrades. It contains a project summary, additional information on the development plan and approach, and a short summary of system support and upgrades.

Joint FRAMES-IWRMS Development Plan

IWRMS will leverage FRAMES (version 2.0) software and extend it in several ways.
The IWRMS development plan calls for several key additions to the current FRAMES system architecture. These activities are summarized below and described in detail in the IWRMS Phase III Statement of Work (see tasks 1, 3, 6, and 7 in the IWRMS Phase III Statement of Work):

- Extensions to the FRAMES Input/Output Dynamic Link Library to capture pedigree information during model runs and interact with a common data repository residing on a machine other than that on which FRAMES is running.
- Development of a Model Integration Wizard to simplify the process of integrating models into the FRAMES environment
- Creation of a Study Manager supporting the partitioning of large analysis efforts into sub-projects and distribution of effort among several team members.
- Distributed computing support, enabling FRAMES to distribute its modeling jobs to a pre-determined pool of computers operating on a Local Area Network (LAN).

In addition, the IWRMS Phase III statement of work calls for developing an external data harvesting tool and the integration of data analysis, GIS, and visualization tools. Both the FRAMES-specific extensions and other development activities may be of value to the current FRAMES funding organizations. The Contractor intends to approach those organizations seeking additional investment in these development activities. If successful, County investments targeted at these developments but not required will be reprogrammed within the IWMRS project to support additional development activities.

Regardless of other agency investment in IWRMS development activities, the project approach calls for integrating (matrixing) the IWRMS and FRAMES development activities such that all organizations using either system benefit in the long run. Specifically, the IWRMS and FRAMES teams will work together to define a common code base (starting with but not limited to the FRAMES 2.0 software) and make changes to that common code base only through mutual agreement between the two efforts.

This requires a development process in which the FRAMES and IWRMS teams will mutually agree upon a common code base and any future changes. Software that is specific to the FRAMES application and not used in IWRMS and that which is specific to IWRMS and not used in the FRAMES application will not be part of this joint development agreement (See Figure 1). The intent is to limit the FRAMES-specific and IWRMS-specific code to the extent possible so as to provide the greatest possible leveraging between these two activities.

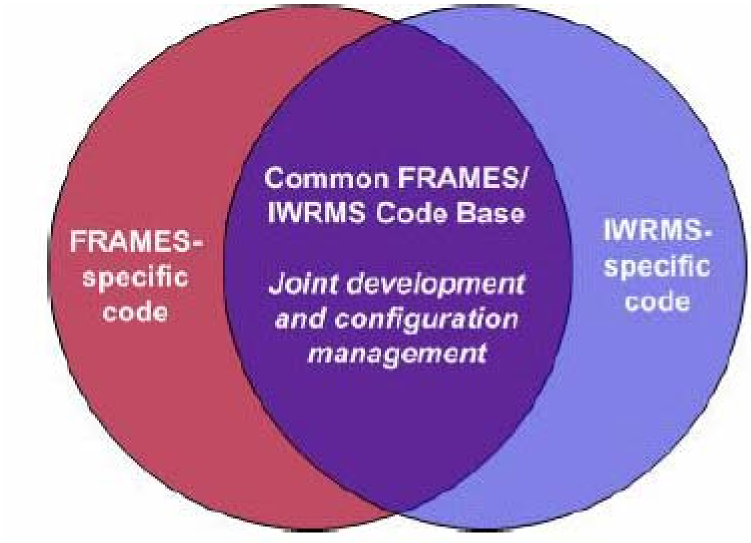


Figure B1 – Code Split Between FRAMES and IWRMS Efforts

The common code base will be carefully managed by personnel from both teams, using industry standard configuration management tools to ensure that changes to the common code base benefit and support both efforts. This approach will require an integration of activities between the IWRMS and FRAMES development teams. Specifically, using personnel from each current team on future activities and associated projects so as to obtain the greatest knowledge transfer between the teams.

IWRMS Long-term Support and Upgrades

Following the completion of IWRMS and delivery to King County, a one-year support contract is provided for in the IWRMS statement of work. During that time, changes to FRAMES and IWRMS will continue to be mutually agreed upon between the two projects, ensuring that changes to the common code base continue to benefit both efforts.

At the conclusion of IWRMS project the following future enhancement conditions will exist:

1. Modifications to the FRAMES/IWRMS code base will be at the sole discretion of the FRAMES development team.
2. To the extent that any future modifications, enhancements, or extensions to the FRAMES/IWRMS common code base are useful to King County, they will be supplied free of charge.
3. Any such changes that require modifications to the IWRMS-specific code base will necessitate King County undertaking such changes on our own and/or contracting with the vendor to accomplish such changes.

Closing

King County will have access to any future upgrades to FRAMES. It will be available in the public domain. However, there is the possibility that future upgrades to FRAMES could be incompatible with modifications made for IWRMS. There is no way to predict that at this time. The goal is for IWRMS and FRAMES to come from the same code base. That is, there will be a

Appendix B

IWRMS and FRAMES
Future Upgrades and Enhancements

set of code that is common to both “products” plus some extensions that are specific to the County's needs and perhaps some extensions that are specific to other FRAMES' clients' needs. However, the goal is to have as much of the software "in common" as possible. Thus, as changes are made to that common software, they will be available to the County as part of the public release nature of the FRAMES software.

Table I. Acronym Definitions

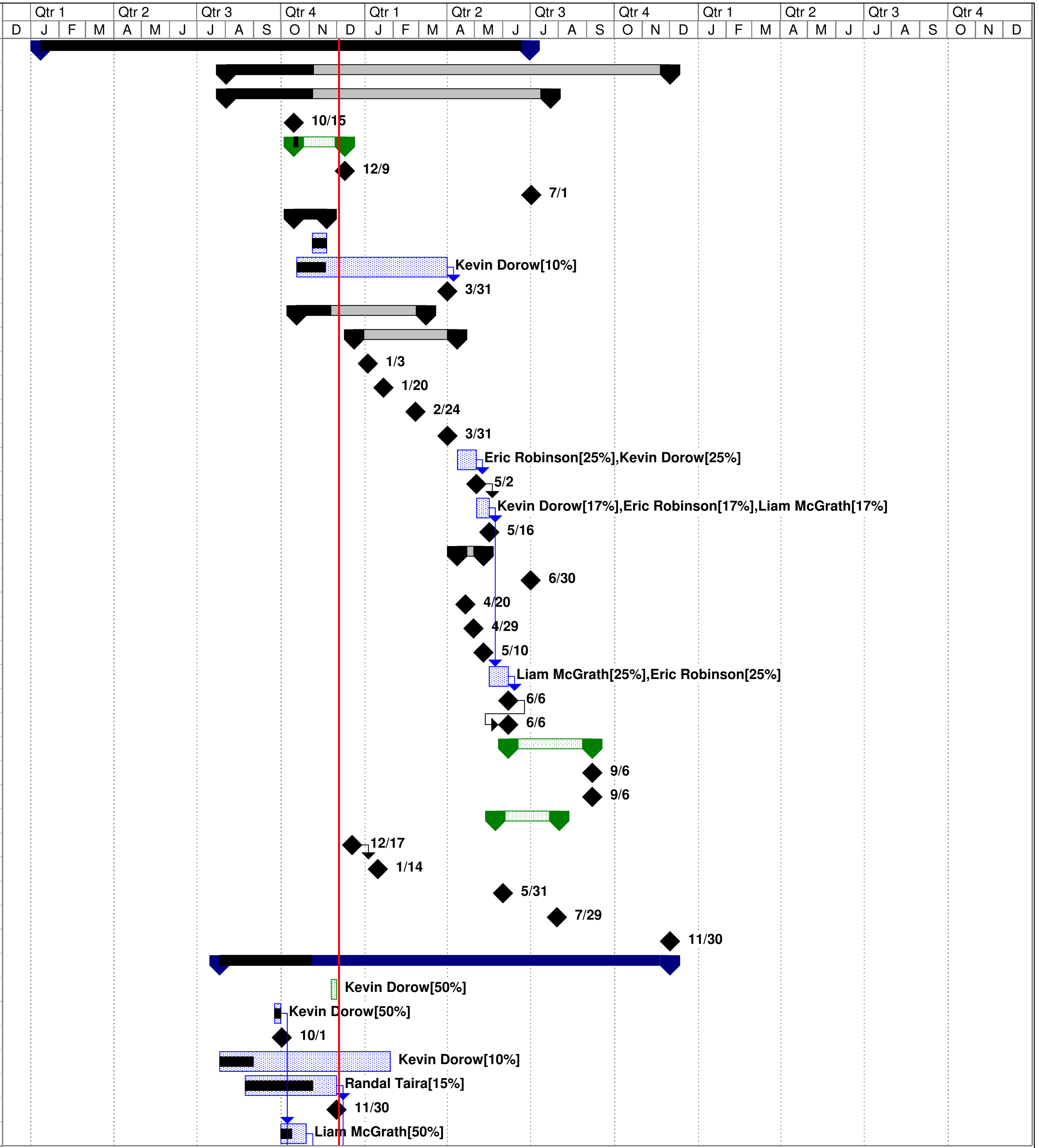
Acronym	Definition
ASCII	American Standard Code for Information Interchange
FRAMES	Framework for Risk Analysis in Multimedia Environmental Systems
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
IWRMS	Integrated Water Resource Modeling System
IODLL	Input/Output Dynamic Link Library
JDBC	Java Data Base Connectivity
KC/DNRP	King County Department of Natural Resources and Parks
ODBC	Open Data Base Connectivity
PNNL	Pacific Northwest National Laboratory
SQL	Standard Query Language
SCR	Software Change Request
UI	User Interface
XML	eXtensible Markup Language

Model: A model in the context of the IWRMS project consists of the executable file, DLL’s, JAR’s, ASCII files, etc, and/or a set of input files for which the executable file is calibrated (or controlled by), a definition of what information within the input files can be changed (acquired during the model registration process) without invalidating the calibration, a set of sample output files, and a definition of how the output files translate into IWRMS defined data specifications (so that relevant information can be passed to other models).

Appendix C

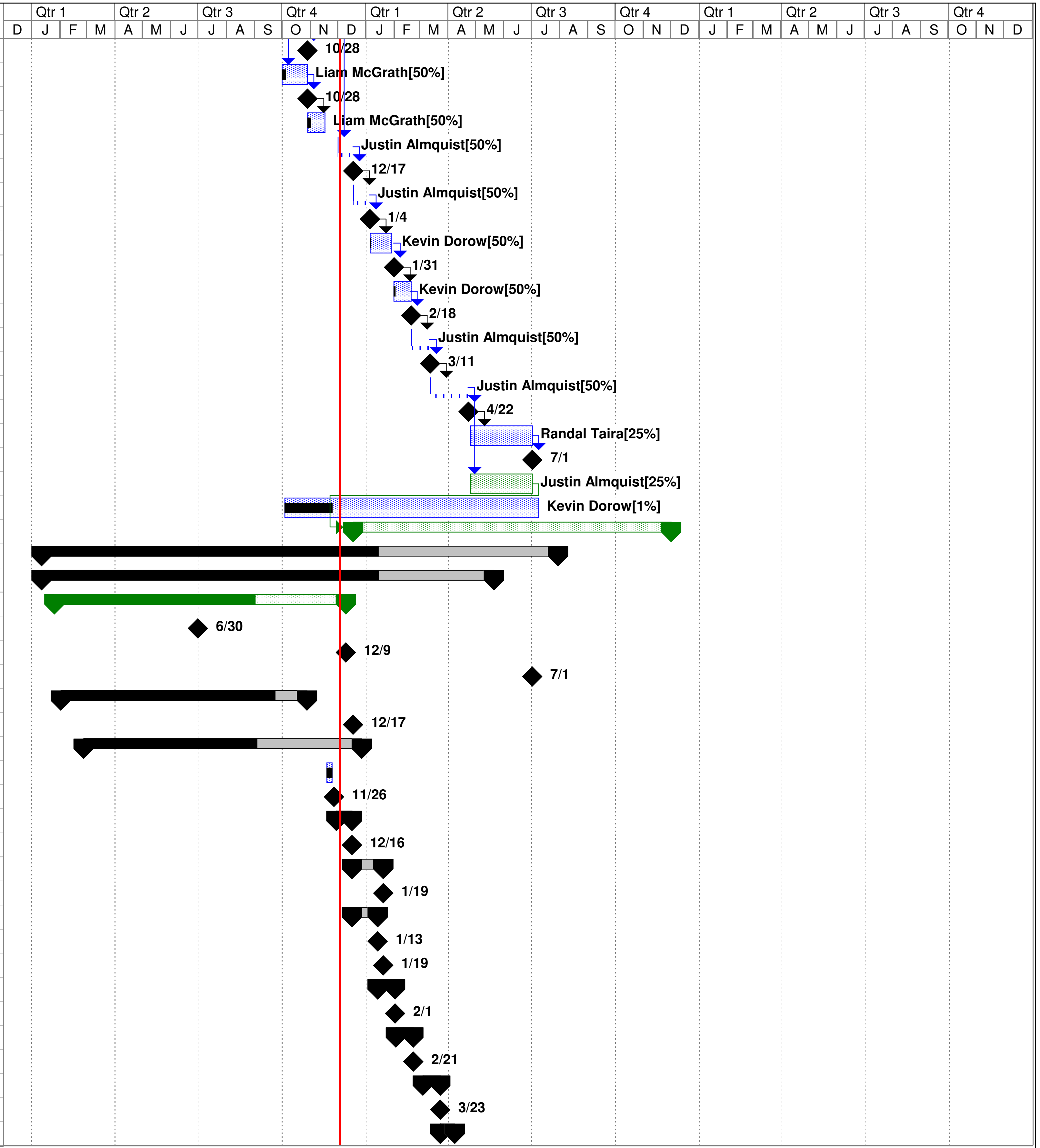
IWRMS Master Schedule

ID	WBS	Task Name	% Complete	Duration	Start	Finish
1	**1**	**Task 1 - FRAMES Baseline Integration**	**98%**	**383 days?**	**Mon 1/12/04**	**Thu 6/30/05**
38	**2**	**Task 2 - External Data Harvester**	**20%**	**348 days**	**Mon 8/2/04**	**Wed 11/30/05**
39	**2.11**	**Task 2 Management**	**27%**	**255 days**	**Mon 8/2/04**	**Fri 7/22/05**
41	2.11	Work Plan Complete	100%	0 days	Fri 10/15/04	Fri 10/15/04
42	**2.1**	**Updated Functional Requirements Specification**	**7%**	**40 days**	**Fri 10/15/04**	**Thu 12/9/04**
44	2.1	Updated Functional Requirements Specification for 12-mont	0%	0 days	Thu 12/9/04	Thu 12/9/04
45	2.1	Updated Functional Requirements Specification for 18-mont	0%	0 days	Fri 7/1/05	Fri 7/1/05
46	**2.2**	**Data Harvester Design**	**100%**	**26 days**	**Fri 10/15/04**	**Fri 11/19/04**
52	2.2	Design Document and Review Complete	100%	12 days	Wed 11/3/04	Fri 11/19/04
53	2.3	Data Harvester Configuration Support	20%	119 days	Mon 10/18/04	Thu 3/31/05
54	2.3	Data Harvester Configuration Support Complete	0%	0 days	Thu 3/31/05	Thu 3/31/05
55	**2.4**	**Data Harvester Administration Development**	**27%**	**101.7 days**	**Mon 10/18/04**	**Tue 3/8/05**
68	**2.5**	**Data Harvester Engine Development**	**1%**	**80.7 days**	**Mon 12/20/04**	**Mon 4/11/05**
81	2.4 & 2.5	Type Database Extractor and UI complete	0%	0 days	Mon 1/3/05	Mon 1/3/05
82	2.4 & 2.5	Type HTTP Extractor, and UI complete	0%	0 days	Thu 1/20/05	Thu 1/20/05
83	2.4 & 2.5	Type FTP/Local Data File Extractor, and UI complete	0%	0 days	Thu 2/24/05	Thu 2/24/05
84	2.4 & 2.5	Type XML Extractor, and UI complete	0%	0 days	Thu 3/31/05	Thu 3/31/05
85	2.6	Create QA/QC database UI	0%	15 days	Mon 4/11/05	Mon 5/2/05
86	2.6	QA/QC Database UI Complete	0%	0 days	Mon 5/2/05	Mon 5/2/05
87	2.7	Modifications to FRAMES Data Owner Tool	0%	10 days	Mon 5/2/05	Mon 5/16/05
88	2.7	Modifications to FRAMES Data Owner Tool Complete	0%	0 days	Mon 5/16/05	Mon 5/16/05
89	**2.9**	**Connect Data Source Types**	**0%**	**20.66 days**	**Mon 4/11/05**	**Tue 5/10/05**
102	2.9.4	Tech Transfer to County for 1 additional Data Source	0%	0 days	Thu 6/30/05	Thu 6/30/05
103	2.9	IWRMS connection to type Database data source complete	0%	0 days	Wed 4/20/05	Wed 4/20/05
104	2.9	IWRMS connection to type HTTP data source complete	0%	0 days	Fri 4/29/05	Fri 4/29/05
105	2.9	IWRMS connection to type FTP/Local Data File data source	0%	0 days	Tue 5/10/05	Tue 5/10/05
106	2.8	Data Harvester Job Scheduler	0%	15 days	Mon 5/16/05	Mon 6/6/05
107	2.8	Data Harvester Scheduler complete	0%	0 days	Mon 6/6/05	Mon 6/6/05
108	2.0	Task 2 - Development Complete	0%	0 days	Mon 6/6/05	Mon 6/6/05
109	**2.10**	**User Documentation**	**0%**	**66 days**	**Mon 6/6/05**	**Tue 9/6/05**
113	2.10	Updated User Documentation Complete	0%	0 days	Tue 9/6/05	Tue 9/6/05
114	2.10	Updated Installation Instructions Complete	0%	0 days	Tue 9/6/05	Tue 9/6/05
115	**10**	**Integration Build and Test Support**	**0%**	**50 days**	**Mon 5/23/05**	**Mon 8/1/05**
119	10	Build and Integration Test Plan for 12-month Demo	0%	0 days	Fri 12/17/04	Fri 12/17/04
120	2	External Data Harvester Functionality Ready for 12-month D	0%	0 days	Fri 1/14/05	Fri 1/14/05
121	10	Build and Integration Test Plan for 18-month Demo	0%	0 days	Tue 5/31/05	Tue 5/31/05
122	2	External Data Harvester Functionality Ready for 18-month	0%	0 days	Fri 7/29/05	Fri 7/29/05
123	2	External Data Harvester to support IWRMS Version 1.0	0%	0 days	Wed 11/30/05	Wed 11/30/05
124	**3**	**Task 3 - Model Integration / Execution Wizard**	**20%**	**353 days?**	**Mon 7/26/04**	**Wed 11/30/05**
125	3.1	Update Functional Requirements Specification	0%	4 days	Thu 11/25/04	Tue 11/30/04
126	3.2	Model Integration Wizard Architecture Review and Design	100%	5 days?	Fri 9/24/04	Thu 9/30/04
127	3.2	Design document for architecture of the Model Integration W	100%	0 days	Fri 10/1/04	Fri 10/1/04
128	3.3	Model Integration Wizard Configuration Support	20%	135 days?	Mon 7/26/04	Fri 1/28/05
129	3.4	County Model Data Specification Development	75%	72 days	Mon 8/23/04	Tue 11/30/04
130	3.4	Data Specifications for all model types to be incorporated in	0%	0 days	Tue 11/30/04	Tue 11/30/04
131	3.5	Model Registration User Interface	40%	20 days?	Fri 10/1/04	Thu 10/28/04



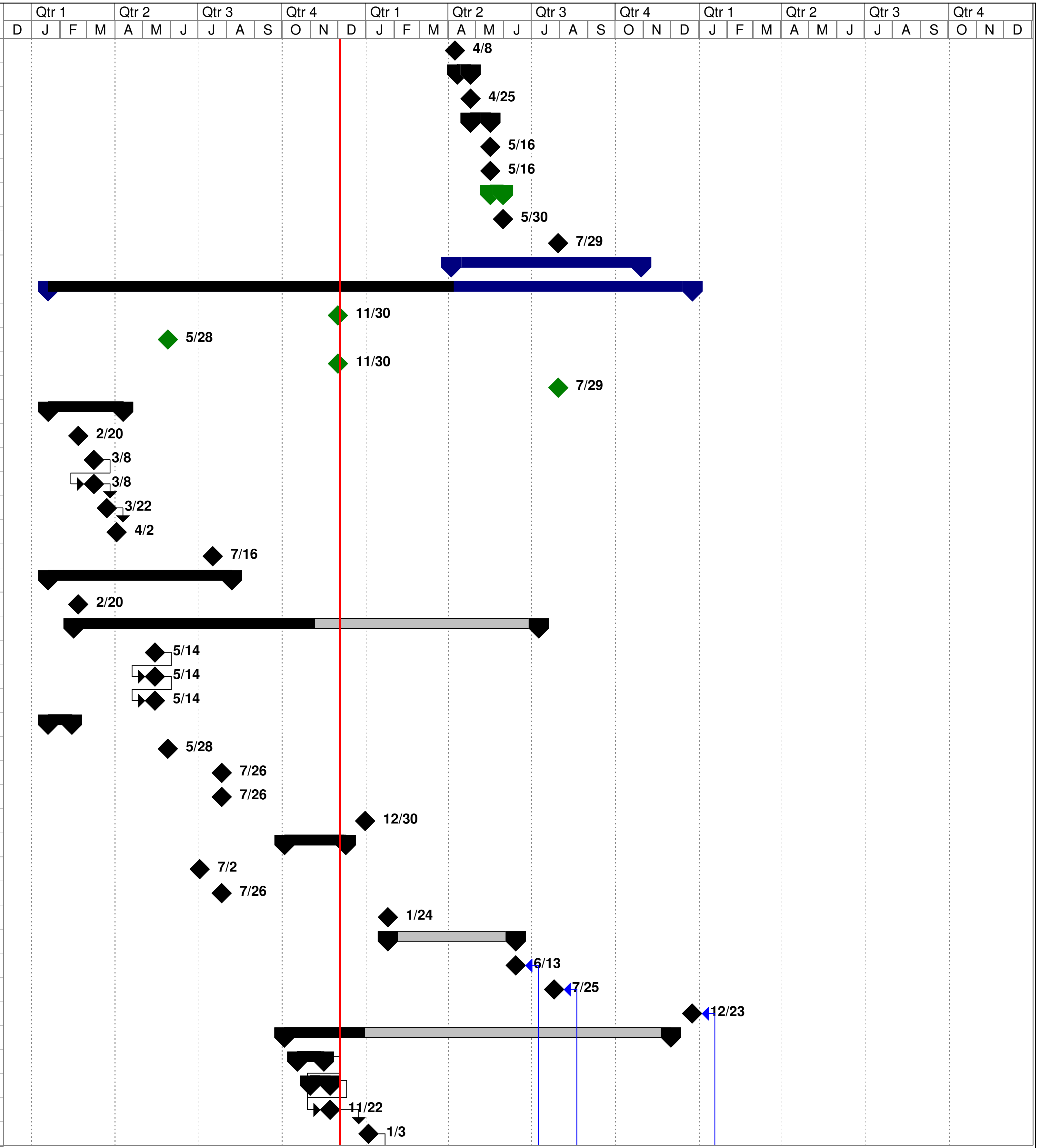
Project: KCConsolidatedMaster-jj-rev1 Date: Fri 12/3/04	Task	▨	Split	┆	Progress	■	Milestone	◆	Summary	▼▬▼	Summary Progress	■

ID	WBS	Task Name	% Complete	Duration	Start	Finish
132	3.5	Generic Model User Interface completed	0%	0 days	Thu 10/28/04	Thu 10/28/04
133	3.6	Control File Definition User Interface	10%	20 days?	Fri 10/1/04	Thu 10/28/04
134	3.6	Control File Definition User Interface completed	0%	0 days	Thu 10/28/04	Thu 10/28/04
135	3.7	Data File Transformation Generator User Interface	10%	13 days?	Fri 10/29/04	Tue 11/16/04
136	3.8	File Input / Output Module	0%	13 days?	Tue 11/30/04	Fri 12/17/04
137	3.8	File Input / Output Module Completed	0%	0 days	Fri 12/17/04	Fri 12/17/04
138	3.9	Parsing Application Programming Interface (API)	0%	12 days?	Fri 12/17/04	Tue 1/4/05
139	3.9	Parsing Application Programming Interface (API) completec	0%	0 days	Tue 1/4/05	Tue 1/4/05
140	3.10	Control File Definition Generator	5%	18.1 days?	Wed 1/5/05	Mon 1/31/05
141	3.10	Control File Definition Generator completed	0%	0 days	Mon 1/31/05	Mon 1/31/05
142	3.11	Data File Transformation Generator	10%	15 days?	Mon 1/31/05	Fri 2/18/05
143	3.11	Data File Definition Generator completed	0%	0 days	Fri 2/18/05	Fri 2/18/05
144	3.12	Data Transformation Engine	0%	15 days?	Fri 2/18/05	Fri 3/11/05
145	3.12	Data Transformation Engine complete	0%	0 days	Fri 3/11/05	Fri 3/11/05
146	3.13	Model Setup Program	0%	30 days?	Fri 3/11/05	Fri 4/22/05
147	3.13	Model Setup Program completed	0%	0 days	Fri 4/22/05	Fri 4/22/05
148	3.14	Integration of County Models	0%	50 days?	Mon 4/25/05	Fri 7/1/05
149	3.14	King County model integration completed	0%	0 days	Fri 7/1/05	Fri 7/1/05
150	3.15	User Documentation	0%	50 days?	Mon 4/25/05	Fri 7/1/05
151	3.16	Task Management	19%	200 days	Mon 10/4/04	Fri 7/8/05
152	**10**	**Integration Build and Test Support**	**0%**	**248 days**	**Fri 12/17/04**	**Wed 11/30/05**
158	**4**	**Task 4 - Data Repository Data Pedigree**	**65%**	**405 days?**	**Mon 1/12/04**	**Fri 7/29/05**
159	**4.13**	**Task 4 Management**	**75%**	**355 days**	**Mon 1/12/04**	**Fri 5/20/05**
163	**4.1**	**Updated Functional Requirements Specification**	**69%**	**229 days**	**Mon 1/26/04**	**Thu 12/9/04**
167	4.1	Updated Functional Requirements Specification for 6-month	100%	0 days	Wed 6/30/04	Wed 6/30/04
168	4.1	Updated Functional Requirements Specification for 12-mont	0%	0 days	Thu 12/9/04	Thu 12/9/04
169	4.1	Updated Functional Requirements Specification for 18-mont	0%	0 days	Fri 7/1/05	Fri 7/1/05
170	**4.5**	**Studies Repository Development**	**87%**	**193.5 days**	**Mon 2/2/04**	**Thu 10/28/04**
179	4.5	Study administration repository development complete	0%	0 days	Fri 12/17/04	Fri 12/17/04
180	**4.2**	**FRAMES Data Repository and Pedigree Integration Deve**	**63%**	**216.5 days?**	**Fri 2/27/04**	**Mon 12/27/04**
204	4.2	Data Repository Administration Design Complete	100%	4 days	Thu 11/18/04	Wed 11/24/04
205	4.2 - 4.4	FRAMES repository development complete	0%	0 days	Fri 11/26/04	Fri 11/26/04
206	**4.11**	**Raw-QA/QC Repository Development**	**2%**	**13 days**	**Mon 11/29/04**	**Thu 12/16/04**
213	4.11	Raw QA/QC Repository Development Complete	0%	0 days	Thu 12/16/04	Thu 12/16/04
214	**4.6**	**Pedigree Repository Development**	**1%**	**24.01 days**	**Thu 12/16/04**	**Wed 1/19/05**
223	4.6	Pedigree repository development complete	0%	0 days	Wed 1/19/05	Wed 1/19/05
224	**4.3**	**Data Specification Repository Development**	**2%**	**20.01 days**	**Thu 12/16/04**	**Thu 1/13/05**
232	4	Data Repository Ready for 12-month Demo	0%	0 days	Thu 1/13/05	Thu 1/13/05
233	4.3	Data specification repository development complete	0%	0 days	Wed 1/19/05	Wed 1/19/05
234	**4.4**	**Data Transformation Repository Development**	**2%**	**13 days**	**Thu 1/13/05**	**Tue 2/1/05**
242	4.4	Data transformation repository development complete	0%	0 days	Tue 2/1/05	Tue 2/1/05
243	**4.7**	**Models Repository Development**	**2%**	**13 days**	**Thu 2/3/05**	**Mon 2/21/05**
250	4.7	Models repository development complete	0%	0 days	Mon 2/21/05	Mon 2/21/05
251	**4.8**	**Datasets Repository Development**	**2%**	**13 days**	**Fri 3/4/05**	**Wed 3/23/05**
258	4.8	Datasets repository development complete	0%	0 days	Wed 3/23/05	Wed 3/23/05
259	**4.10**	**Applications Repository Development**	**2%**	**11.75 days**	**Wed 3/23/05**	**Fri 4/8/05**



Project: KCConsolidatedMaster-jj-rev1 Date: Fri 12/3/04	Task ▨	Split ⋯⋯	Progress ▬	Milestone ◆	Summary	Summary Progress ▬

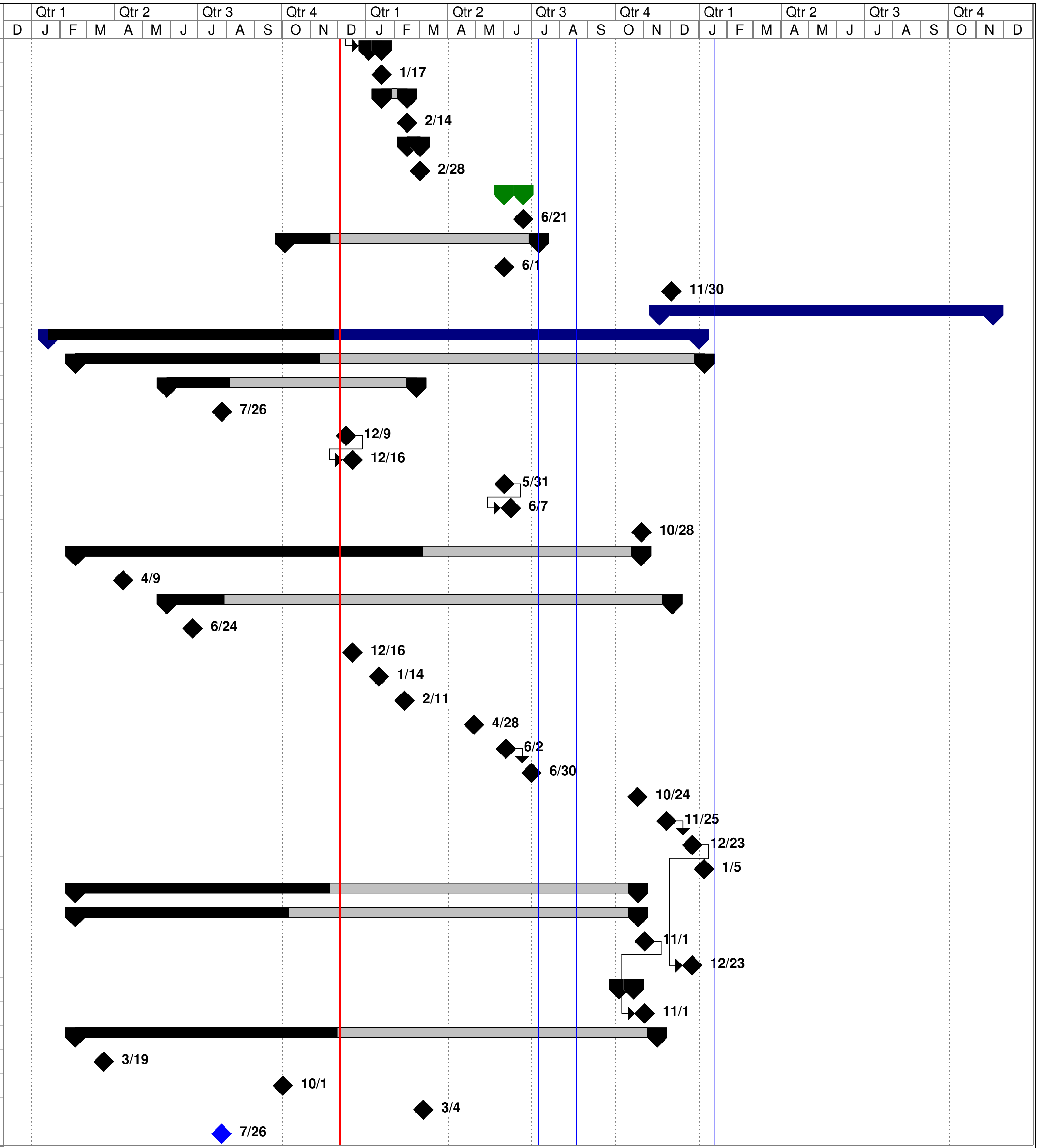
ID	WBS	Task Name	% Complete	Duration	Start	Finish
266	4.10	Applications repository development complete	0%	0 days	Fri 4/8/05	Fri 4/8/05
267	**4.9**	**Reports Repository Development**	**2%**	**10.5 days**	**Mon 4/11/05**	**Mon 4/25/05**
274	4.9	Reports repository development complete	0%	0 days	Mon 4/25/05	Mon 4/25/05
275	**4.12**	**Archiver Repository Development**	**1%**	**15.5 days**	**Mon 4/25/05**	**Mon 5/16/05**
282	4.12	Data Repository Ready for 18-month Demo	0%	0 days	Mon 5/16/05	Mon 5/16/05
283	4.12	Archiver repository development complete	0%	0 days	Mon 5/16/05	Mon 5/16/05
284	**10**	**Integration Build and Test**	**0%**	**10 days**	**Tue 5/17/05**	**Mon 5/30/05**
286	4	Task 4 Development Complete	0%	0 days	Mon 5/30/05	Mon 5/30/05
287	4	Updates to the repository and data model as needed to sup	0%	0 days	Fri 7/29/05	Fri 7/29/05
288	**5**	**Task 5 - Visual Analysis Tool Integration**	**0%**	**150 days**	**Mon 4/4/05**	**Fri 10/28/05**
290	**6**	**Task 6 - Study Manager UI**	**63%**	**505 days**	**Mon 1/19/04**	**Fri 12/23/05**
291	**6.1**	**Update Functional Requirements Specification**	**0%**	**0 days**	**Tue 11/30/04**	**Tue 11/30/04**
293	6.1	Update Functional Requirements Specification 6-month	100%	0 days	Fri 5/28/04	Fri 5/28/04
294	6.1	Update Functional Requirements Specification 12-month	0%	0 days	Tue 11/30/04	Tue 11/30/04
295	6.1	Update Functional Requirements Specification 18-month	0%	0 days	Fri 7/29/05	Fri 7/29/05
296	**6.5**	**Refine Paper Prototype Design**	**100%**	**60 days**	**Mon 1/19/04**	**Fri 4/9/04**
305	6	Complete review of current UI design	100%	0 days	Fri 2/20/04	Fri 2/20/04
306	6	Draft paper prototype design document	100%	0 days	Mon 3/8/04	Mon 3/8/04
307	6	Draft design document submitted to the County	100%	0 days	Mon 3/8/04	Mon 3/8/04
308	6	[County Deliverable to PNNL] Comments from County on dr	100%	0 days	Mon 3/22/04	Mon 3/22/04
309	6	Updated design document complete	100%	0 days	Fri 4/2/04	Fri 4/2/04
310	6.3 - 6.6	Updated design document complete	100%	0 days	Fri 7/16/04	Fri 7/16/04
311	**6.3**	**Set-up Development Environment**	**100%**	**145 days**	**Mon 1/19/04**	**Fri 8/6/04**
320	6	Set up development environment complete	100%	0 days	Fri 2/20/04	Fri 2/20/04
321	**6.4**	**Implement User Interface**	**52%**	**365 days**	**Mon 2/16/04**	**Fri 7/8/05**
355	6.4	Study manager component prototype	100%	0 days	Fri 5/14/04	Fri 5/14/04
356	6.4	New study wizard component prototype	100%	0 days	Fri 5/14/04	Fri 5/14/04
357	6.4	New study wizard component prototype	100%	0 days	Fri 5/14/04	Fri 5/14/04
358	**6.5**	**Incorporate Data**	**100%**	**20 days**	**Mon 1/19/04**	**Fri 2/13/04**
361	6.5	Combined prototype	100%	0 days	Fri 5/28/04	Fri 5/28/04
362	6.5	Updated combined prototype for 6-month demo	100%	0 days	Mon 7/26/04	Mon 7/26/04
363	6.3 - 6.5	Study manager to support 6-month demo	100%	0 days	Mon 7/26/04	Mon 7/26/04
364	6.3 - 6.5	Study manager to support 12-month demo	0%	0 days	Thu 12/30/04	Thu 12/30/04
365	**6.6**	**Conduct Usability Studies**	**95%**	**49 days**	**Mon 10/4/04**	**Thu 12/9/04**
369	6.6	Internal usability tests complete	100%	0 days	Fri 7/2/04	Fri 7/2/04
370	6.6	External usability tests	100%	0 days	Mon 7/26/04	Mon 7/26/04
371	6.6	External usability tests conducted at 12-month demonstratic	0%	0 days	Mon 1/24/05	Mon 1/24/05
372	**6.7**	**IWRMS Integration**	**0%**	**100 days**	**Tue 1/25/05**	**Mon 6/13/05**
374	6.7	Integration with Rest of IWRMS System	0%	0 days	Mon 6/13/05	Mon 6/13/05
375	6.7	Study Manager to Support 18 Month System	0%	0 days	Mon 7/25/05	Mon 7/25/05
376	6.7	Study Manager to Support IWRMS Version 1.0	0%	0 days	Fri 12/23/05	Fri 12/23/05
377	**7**	**Task 7 - Distributed Computing**	**21%**	**303 days**	**Mon 10/4/04**	**Wed 11/30/05**
378	**7.1**	**Determine Hardware for Dedicated Computing Cluster**	**95%**	**21 days**	**Mon 10/18/04**	**Mon 11/15/04**
380	**7.2**	**Determine Software for Managing the Dedicated Compu**	**50%**	**16 days**	**Mon 11/1/04**	**Mon 11/22/04**
382	7	Determine Hardware / Software for Dedicated Computing Cl	0%	0 days	Mon 11/22/04	Mon 11/22/04
383	7	Order Software/Hardware	0%	0 days	Mon 1/3/05	Mon 1/3/05



Project: KCConsolidatedMaster-jj-rev1
Date: Fri 12/3/04

Task	▨	Split	┆	Progress	■	Milestone	◆	Summary	▬	Summary Progress	■

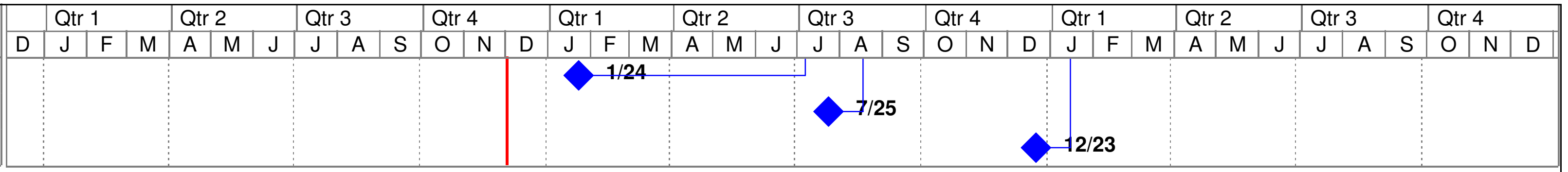
ID	WBS	Task Name	% Complete	Duration	Start	Finish
384	**7.3**	**Install & Configure Computing Cluster**	**0%**	**10 days**	**Tue 1/4/05**	**Mon 1/17/05**
387	7.3	Install and Configure the Computing Cluster Hardware / Sof	0%	0 days	Mon 1/17/05	Mon 1/17/05
388	**7.4**	**Integrate the Dedicated Computing Cluster Functionalit**	**0%**	**20 days**	**Tue 1/18/05**	**Mon 2/14/05**
390	7.4	Integrate the Computing Cluster into the IWRMS	0%	0 days	Mon 2/14/05	Mon 2/14/05
391	**7.5**	**Testing Computer Cluster**	**0%**	**10 days**	**Tue 2/15/05**	**Mon 2/28/05**
393	7.5	Test the Dedicated Computing Cluster	0%	0 days	Mon 2/28/05	Mon 2/28/05
394	**7.6**	**User Documentation**	**0%**	**15 days**	**Wed 6/1/05**	**Tue 6/21/05**
396	7.6	Dedicated Computing Cluster Documentation	0%	0 days	Tue 6/21/05	Tue 6/21/05
397	**7.7**	**Task Management**	**18%**	**200 days**	**Mon 10/4/04**	**Fri 7/8/05**
399	7	Dedicated Computing Cluster functionality to support 18 mo	0%	0 days	Wed 6/1/05	Wed 6/1/05
400	7	Dedicated Computing Cluster to support IWRMS Version 1.	0%	0 days	Wed 11/30/05	Wed 11/30/05
401	**8**	**Task 8 - Long Term Support**	**0%**	**261 days**	**Fri 11/18/05**	**Fri 11/17/06**
403	**9**	**Task 9 - Project Management**	**44%**	**510 days**	**Mon 1/19/04**	**Fri 12/30/05**
405	**10**	**Task 10 Implementation Support**	**39%**	**492 days?**	**Wed 2/18/04**	**Thu 1/5/06**
406	**10.1**	**Requirements Management**	**25%**	**195 days**	**Fri 5/28/04**	**Thu 2/24/05**
435	10.1	6-month Deliverable Requirements To County	100%	0 days	Mon 7/26/04	Mon 7/26/04
436	10.1	12-month Deliverable Requirements To County	0%	0 days	Thu 12/9/04	Thu 12/9/04
437	10.1	[County Deliverable to PNNL] County review of 12-month	0%	0 days	Thu 12/16/04	Thu 12/16/04
438	10.1	18-month Deliverable Requirements To County	0%	0 days	Tue 5/31/05	Tue 5/31/05
439	10.1	[County Deliverable to PNNL] County review of 18-month	0%	0 days	Tue 6/7/05	Tue 6/7/05
440	10.1	IWRMS Version 1 Deliverable Requirements To County	0%	0 days	Fri 10/28/05	Fri 10/28/05
441	**10.2**	**Configuration Management Activities**	**61%**	**443 days?**	**Wed 2/18/04**	**Fri 10/28/05**
472	10.2	Configuration Management Plan Complete	100%	0 days	Fri 4/9/04	Fri 4/9/04
473	**10.3**	**Integration and System Testing Activities**	**11%**	**395 days**	**Fri 5/28/04**	**Thu 12/1/05**
493	10.3	Update FRAMES test suite to reflect changes	100%	0 days	Thu 6/24/04	Thu 6/24/04
494	10.3	IWRMS test plan complete (12-month demonstration)	0%	0 days	Thu 12/16/04	Thu 12/16/04
495	10.3	Build and test (12-month demonstration)	0%	0 days	Fri 1/14/05	Fri 1/14/05
496	10.3	IWRMS test report complete (12-month system)	0%	0 days	Fri 2/11/05	Fri 2/11/05
497	10.3	IWRMS test plan complete (18-month demonstration)	0%	0 days	Thu 4/28/05	Thu 4/28/05
498	10.3	Build and test (18-month demonstration)	0%	0 days	Thu 6/2/05	Thu 6/2/05
499	10.3	IWRMS test report complete (18-month system)	0%	0 days	Thu 6/30/05	Thu 6/30/05
500	10.3	IWRMS Beta Version software system	0%	0 days	Mon 10/24/05	Mon 10/24/05
501	10.3	Build and testing (IWRMS Version 1.0)	0%	0 days	Fri 11/25/05	Fri 11/25/05
502	10.3	IWRMS Version 1.0 software system complete	0%	0 days	Fri 12/23/05	Fri 12/23/05
503	10.3	IWRMS test report complete (IWRMS Version 1.0)	0%	0 days	Thu 1/5/06	Thu 1/5/06
504	**10.4**	**System Administration Support**	**45%**	**440 days**	**Wed 2/18/04**	**Tue 10/25/05**
507	**10.5**	**Documentation**	**38%**	**440 days**	**Wed 2/18/04**	**Tue 10/25/05**
533	10.5	IWRMS draft documentation complete	0%	0 days	Tue 11/1/05	Tue 11/1/05
534	10.5	IWRMS Version 1.0 documentation complete	0%	0 days	Fri 12/23/05	Fri 12/23/05
535	**10.6**	**Training and Support**	**0%**	**12 days**	**Wed 10/5/05**	**Thu 10/20/05**
538	10.6	Training Complete	0%	0 days	Tue 11/1/05	Tue 11/1/05
539	**10.7**	**Task 10 Management**	**45%**	**455 days**	**Wed 2/18/04**	**Tue 11/15/05**
541	10.7	Microsoft Project schedule (6- month demonstration)	100%	0 days	Fri 3/19/04	Fri 3/19/04
542	10.7	Microsoft Project schedule (12- month demonstration)	100%	0 days	Fri 10/1/04	Fri 10/1/04
543	10.7	Microsoft Project schedule (18- month demonstration)	0%	0 days	Fri 3/4/05	Fri 3/4/05
544	11	Six Month Prototype Demonstration	100%	0 days	Mon 7/26/04	Mon 7/26/04



Project: KCConsolidatedMaster-jj-rev1
Date: Fri 12/3/04

Task	▨	Split	⋯	Progress	■	Milestone	◆	Summary		Summary Progress	■

ID	WBS	Task Name	% Complete	Duration	Start	Finish
545	12	Twelve Month Prototype Demonstration	0%	0 days	Mon 1/24/05	Mon 1/24/05
546	13	Initial System (18 months)	0%	0 days	Mon 7/25/05	Mon 7/25/05
547	14	Final System Delivery	0%	0 days	Fri 12/23/05	Fri 12/23/05


Qtr 1
Qtr 2
Qtr 3
Qtr 4
Qtr 1
Qtr 2
Qtr 3
Qtr 4
Qtr 1
Qtr 2
Qtr 3
Qtr 4
D J F M A M J J A S O N D J F M A M J J A S O N D J F M A M J J A S O N D
1/24
7/25
12/23